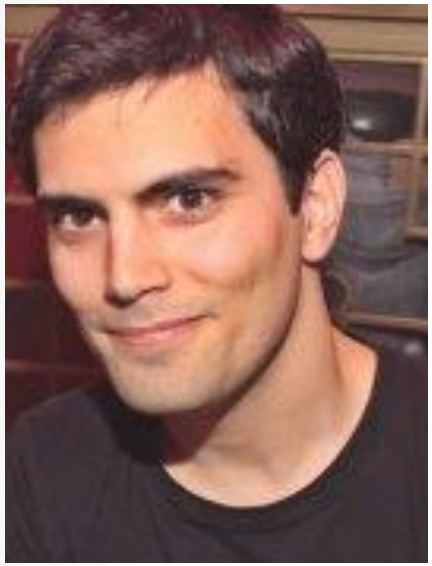
PROV-O: The PROV Ontology Tutorial

Daniel Garijo

Ontology Engineering Group
Universidad Politécnica de Madrid

(with Slides from Luc Moreau, Ivan Herman, Paul Groth and Timothy Lebo)

Date: 01/09/2013



Daniel Garijo
PhD student at the
Ontology Engineering Group
Universidad Politécnica de Madrid
(http://delicias.dia.fi.upm.es/members/dgarijo/)

Member of:
W3C Provenance Incubator Group
W3C Provenance Working Group
DCMI Metadata Provenance Task Group

Download this presentation here:
http://www.slideshare.net/dgarijo/provo-tutorial-dc2013-conference

Plan for the afternoon

- Introduction: The Provenance Working Group
- From DC to PROV: An example
- PROV-O: An Overview
- PROV-O part 1
- Coffee Break
- PROV-O part 2
- Mapping PROV-O to Dublin Core

Introduction: The Provenance Working Group

- We aim to express how data has evolved:
  - Who played what role in creating the data.
  - Who owned the data.
  - Who contributed to the data.
  - How data was modified from its first revision.
  - How other data affected the current data.
  - What tools where used to generate each version of the data
  - etc.

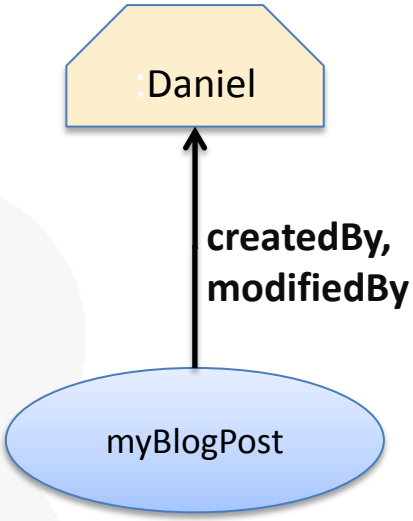
Image from : http://www.psmag.com/science/the-background-on-your-bytes-40220/

…but it is not that easy!

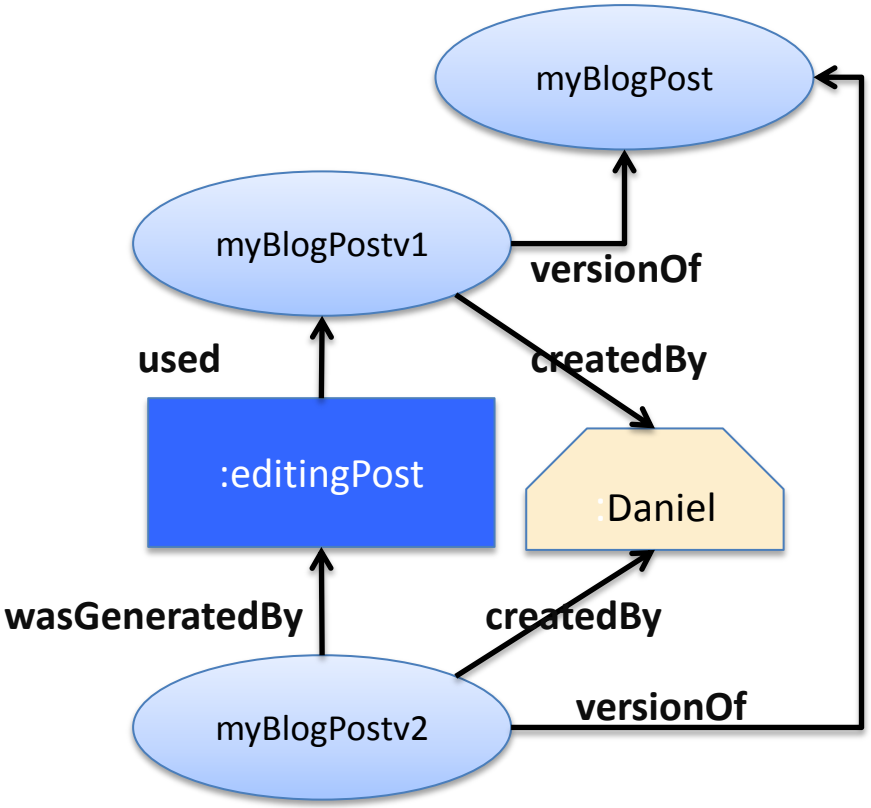
- Requires a complete model describing the various constituents (actors, revisions, etc.)
- The model should be usable with RDF to be used on the Semantic Web
- Has to find a balance between provenance granularities
  - simple (“scruffy”) provenance: easily usable.
  - complex (“complete”) provenance: allows for a detailed reporting of origins, versions, etc.

Example: Editing a blog post

Scruffy provenance



Complete provenance



Applications of provenance

- Art
  - Ownership of pieces of art
- Open Information Systems
  - origin of the data, who was responsible for its creation
- Science applications
  - how the results of a publication were obtained (scientific workflows)
- News
  - origins and references of blogs, news items
- Law
  - licensing attribution of documents, data
  - privacy information
- Etc.

- *"Provenance is defined as a record that describes the people, institutions, entities, and activities involved in producing, influencing, or delivering a piece of data or a thing".*

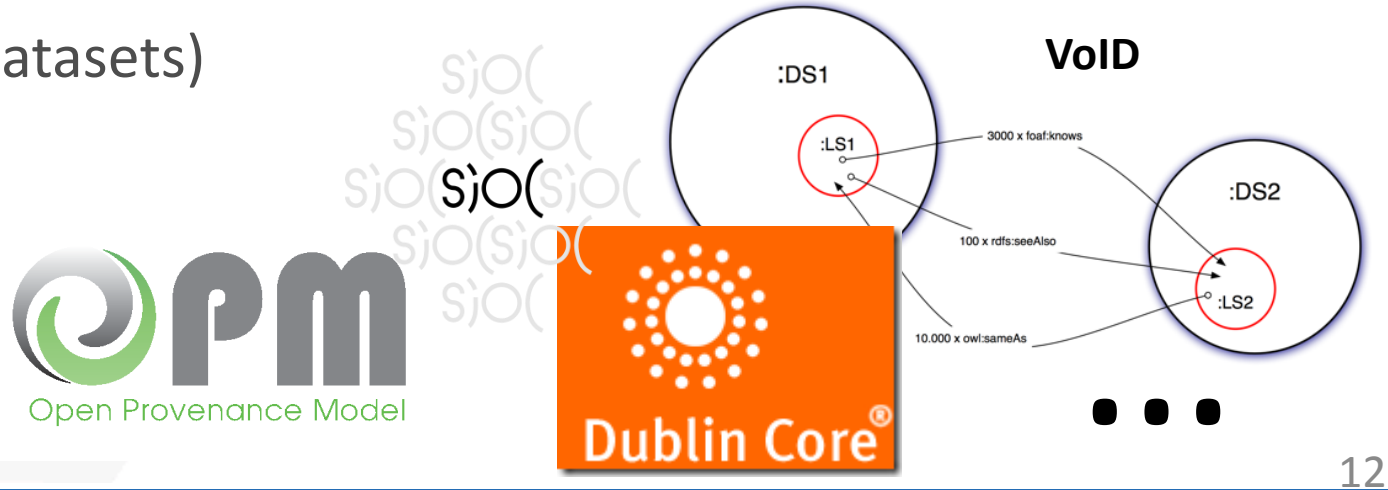
W3C Provenance Incubator Group

- *"Provenance assertions are a form of contextual metadata and can themselves become important records with their own provenance."*

- Provenance is Metadata, but not all metadata is provenance:
  - The title or format of a book is metadata, but it not part of its provenance.
  - The date of creation, the author, the publisher or the license of a book are part of its provenance.

- A lot of work has been done in
  - Workflow management systems
    - Reproducibility, repeatability and attribution
  - Databases
    - Who modified a record? How?
  - knowledge representation
    - How was an entity affected?
  - information retrieval
    - Who is responsible for this information?

- Communities and vocabularies are already in use:
  - Dublin Core (documents and resources)
  - Open Provenance Model (OPM) and extensions
  - Provenir ontology (sensor networks)
  - Provenance vocabulary (Linked Data)
  - SWAN provenance ontology (Neuromedicine resources)
  - SIOC (online blogs and forums)
  - VOID (datasets)
  - etc.



- The existing models track provenance at different granularities in different domains.
  - How do we make the provenance descriptions interchangeable?
  - How do we integrate these heterogeneous provenance data?

- Worked in 2009-2010 (Chaired by Yolanda Gil)
- Issued a final report
  - "Provenance XG Final Report"
    - http://www.w3.org/2005/Incubator/prov/XGR-prov/
  - provides an overview of the various existing approaches and vocabularies
  - proposes the creation of a dedicated W3C Working Group
    - A set of terms is recommended for initial discussion

- Introduces requirements for the provenance in the web: http://www.w3.org/2005/Incubator/prov/wiki/User_Requirements
- Maps different existing vocabulary approaches to OPM: http://www.w3.org/2005/Incubator/prov/wiki/Provenance_Vocabulary_Mappings
- Defines three common use case scenarios for provenance
  - News Aggregator
  - Disease Outbreak
  - Business Contract

- Set up in April 2011
  - (co-chaired by Paul Groth and Luc Moreau)

- Goal is to define a standard way to interchange provenance on the web.

- Focused on the Semantic Web.

Participants

- DERI Galway
- European Broadcasting Union
- FORTH
- Financial Services Technology Consortium
- DFKI
- IBBT
- IBM
- Library of Congress
- Mayo Clinic
- NASA
- OCLC
- Open Geospatial Consortium
- OpenLink Software
- Oracle
- Pacific Northwest National Laboratory
- Rensselaer Polytechnic Institute
- Revelytix, Inc
- Newcastle University
- The National Archives
- TopQuadrant
- Universidad Politécnica de Madrid
- University of Aberdeen
- University of Edinburgh
- University of Manchester
- University of Oxford
- University of Southampton
- VU University Amsterdam
- Wright State University

+ Invited Experts

- Main documents:
  - PROV Overview (http://www.w3.org/TR/prov-overview/)
  - PROV Primer (http://www.w3.org/TR/prov-primer/)
  - PROV Data Model(*) (http://www.w3.org/TR/prov-dm/)
  - PROV Constraints(*) (http://www.w3.org/TR/prov-constraints/)
  - PROV Semantics (http://www.w3.org/TR/prov-sem/)
  - PROV Notation(*) (http://www.w3.org/TR/prov-n/)
  - PROV Ontology(*) (http://www.w3.org/TR/prov-o/)
  - PROV XML Serialization (http://www.w3.org/TR/prov-xml/)
  - PROV Access and Query (http://www.w3.org/TR/prov-aq/)
  - PROV DC Mapping (http://www.w3.org/TR/prov-dc/)
  - PROV Links (http://www.w3.org/TR/prov-links/)
  - PROV Dictionary (http://www.w3.org/TR/prov-dictionary/)
  - PROV Implementations (http://www.w3.org/TR/prov-implementations/)

(*)Rec-track documents

- Main documents:
  - PROV Overview (http://www.w3.org/TR/prov-overview/)
  - PROV Primer (http://www.w3.org/TR/prov-primer/)
  - PROV Data Model(*) (http://www.w3.org/TR/prov-dm/)
  - PROV Constraints(*) (http://www.w3.org/TR/prov-constraints/)
  - PROV Semantics (http://www.w3.org/TR/prov-sem/)

  Model

  - PROV Notation(*) (http://www.w3.org/TR/prov-n/)
  - PROV Ontology(*) (http://www.w3.org/TR/prov-o/)
  - PROV XML Serialization (http://www.w3.org/TR/prov-xml/)

  Serializations

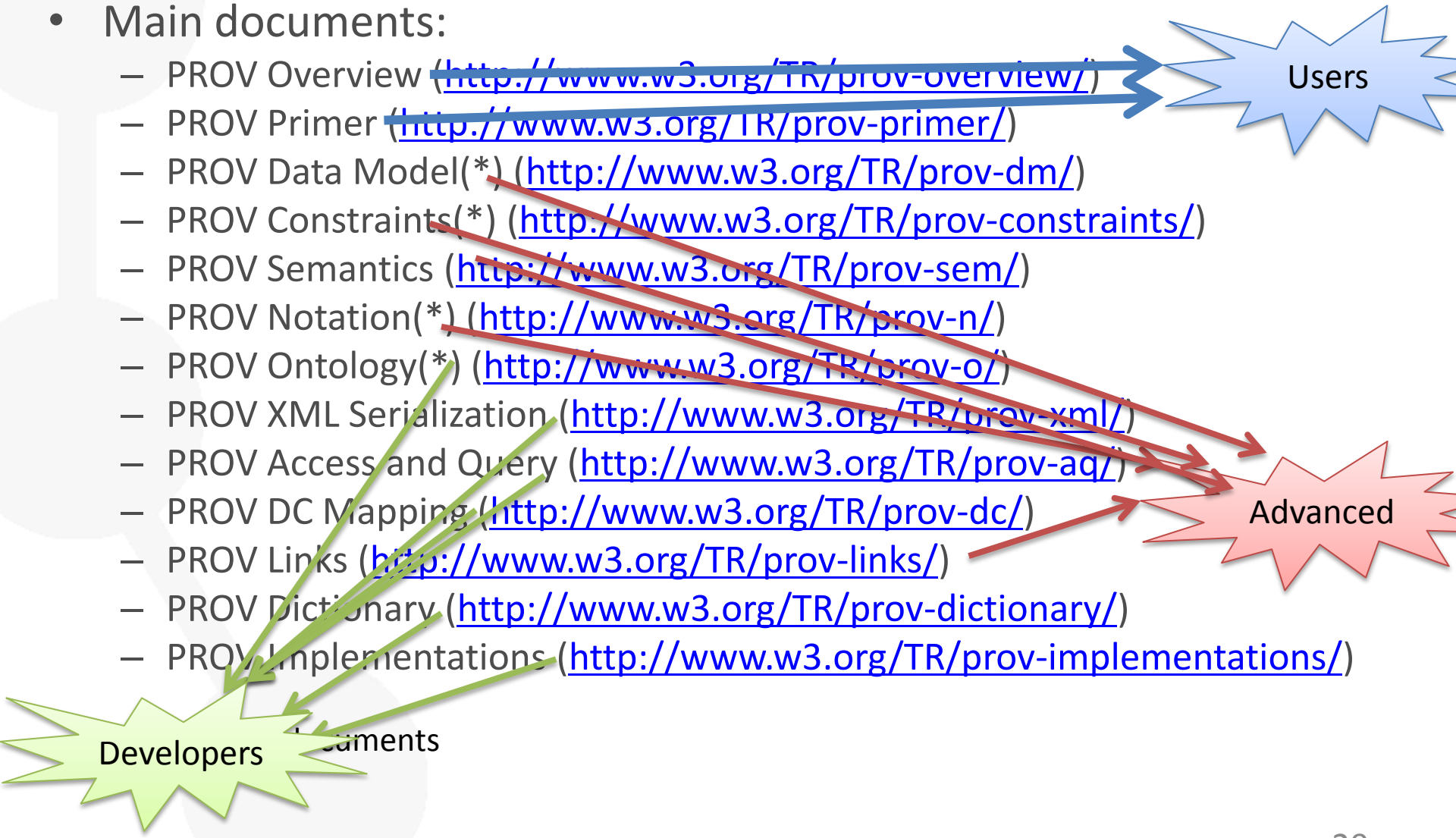
  - PROV Access and Query (http://www.w3.org/TR/prov-aq/)
  - PROV DC Mapping (http://www.w3.org/TR/prov-dc/)
  - PROV Links (http://www.w3.org/TR/prov-links/)
  - PROV Dictionary (http://www.w3.org/TR/prov-dictionary/)

  Extensions

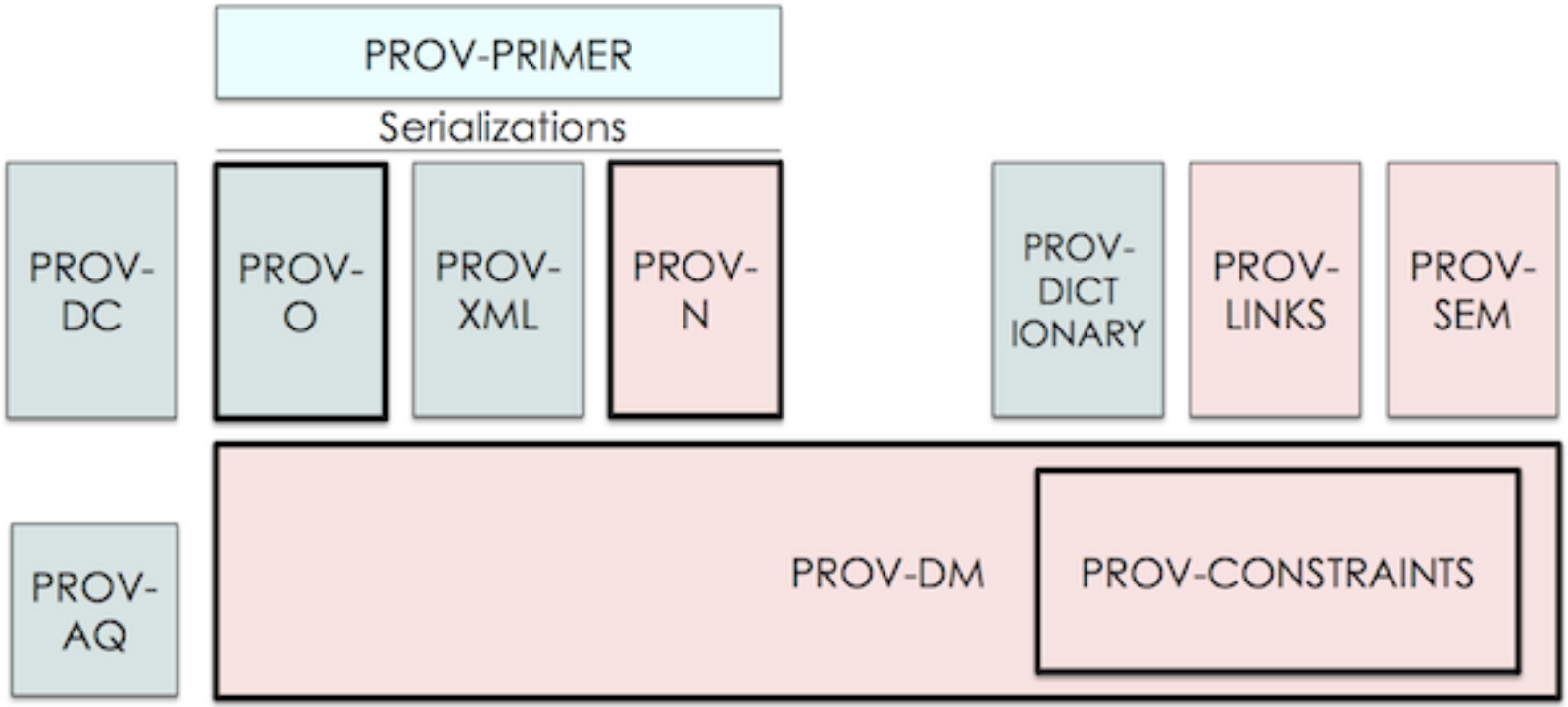
  - PROV Implementations (http://www.w3.org/TR/prov-implementations/)

(*)Rec-track documents

- Main documents:
  - PROV Overview (http://www.w3.org/TR/prov-overview/)
  - PROV Primer (http://www.w3.org/TR/prov-primer/)
  - PROV Data Model(*) (http://www.w3.org/TR/prov-dm/)
  - PROV Constraints(*) (http://www.w3.org/TR/prov-constraints/)
  - PROV Semantics (http://www.w3.org/TR/prov-sem/)
  - PROV Notation(*) (http://www.w3.org/TR/prov-n/)
  - PROV Ontology(*) (http://www.w3.org/TR/prov-o/)
  - PROV XML Serialization (http://www.w3.org/TR/prov-xml/)
  - PROV Access and Query (http://www.w3.org/TR/prov-aq/)
  - PROV DC Mapping (http://www.w3.org/TR/prov-dc/)
  - PROV Links (http://www.w3.org/TR/prov-links/)
  - PROV Dictionary (http://www.w3.org/TR/prov-dictionary/)
  - PROV Implementations (http://www.w3.org/TR/prov-implementations/)



PROV dependencies



- The Rec Track documents have been released as CR (candidate recommendations)

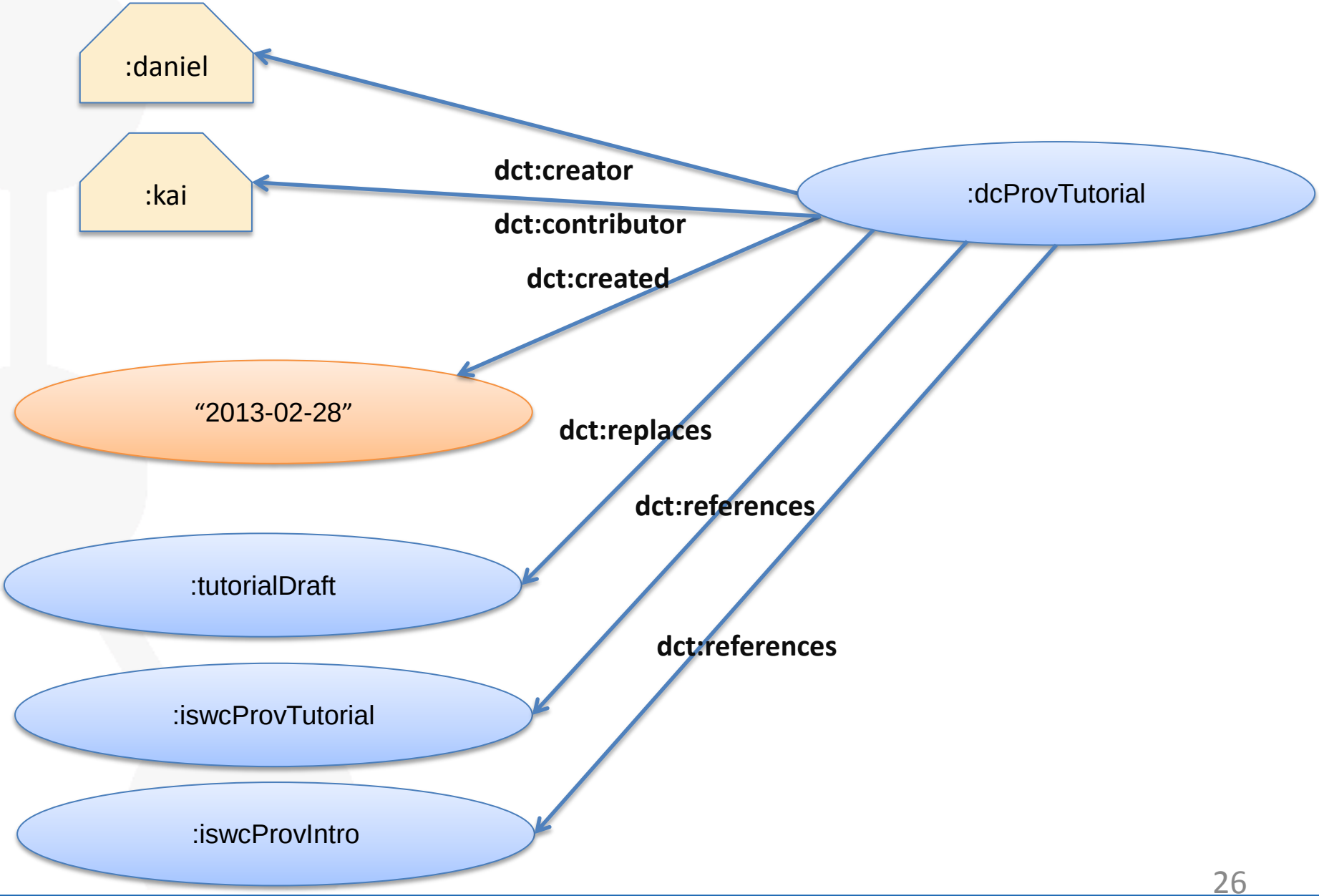
- Group has finished:
  - Official Wiki (frozen): http://www.w3.org/2011/prov/wiki/Main_Page
  - Active Semantic Wiki: http://www.w3.org/2001/sw/wiki/PROV
  - FAQ: http://www.w3.org/2001/sw/wiki/PROV-FAQ

From Dublin Core to PROV-O
An example

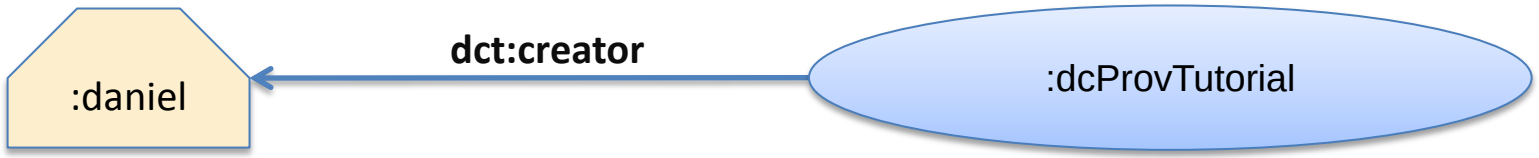
- This presentation
  - Created by Daniel.
  - Kai contributed with feedback.
  - Used previous tutorials as references.
  - Refinement of previous presentation draft.

```
:dcProvTutorial a foaf:Document;
        dct:title “PROV-O Tutorial" ;
        dct:creator :daniel ;
        dct:contributor :kai;
        dct:created "2013-08-25" ;
        dct:replaces :tutorialDraft;
        dct:references :iswcProvTutorial, :iswcProvIntro.

:kai a dct:Agent.

:daniel a dct:Agent.

:tutorialDraft a foaf:Document;
        dct:creator :daniel.

:iswcProvTutorial a foaf:Document;
...
```

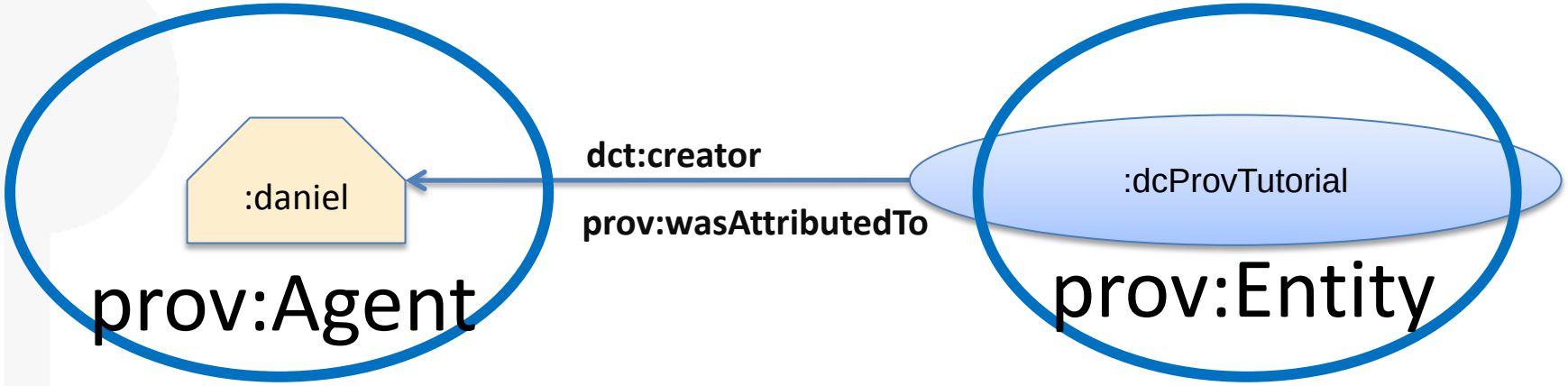
Graphical representation

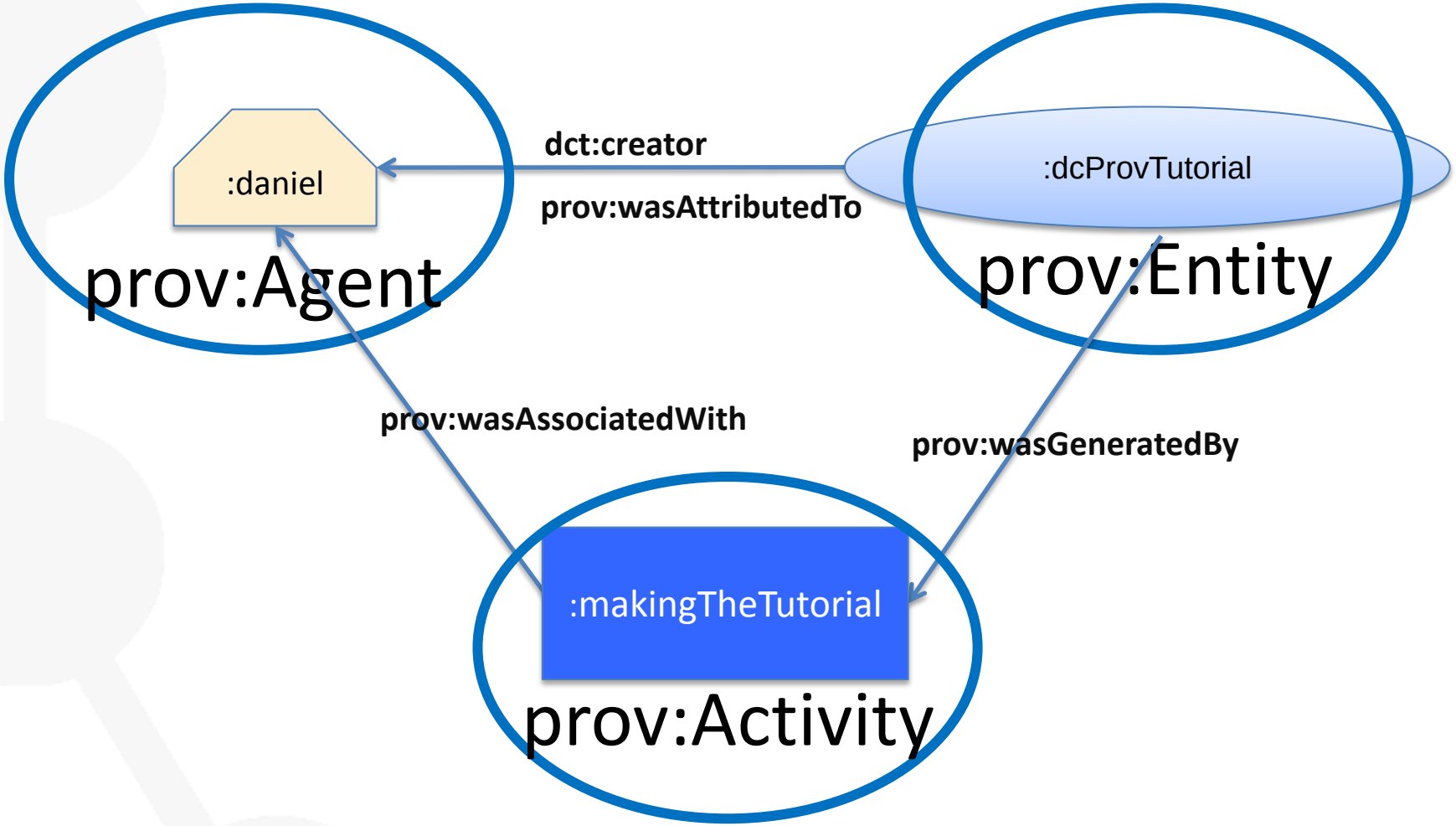


PROV-O: A very simple attribution

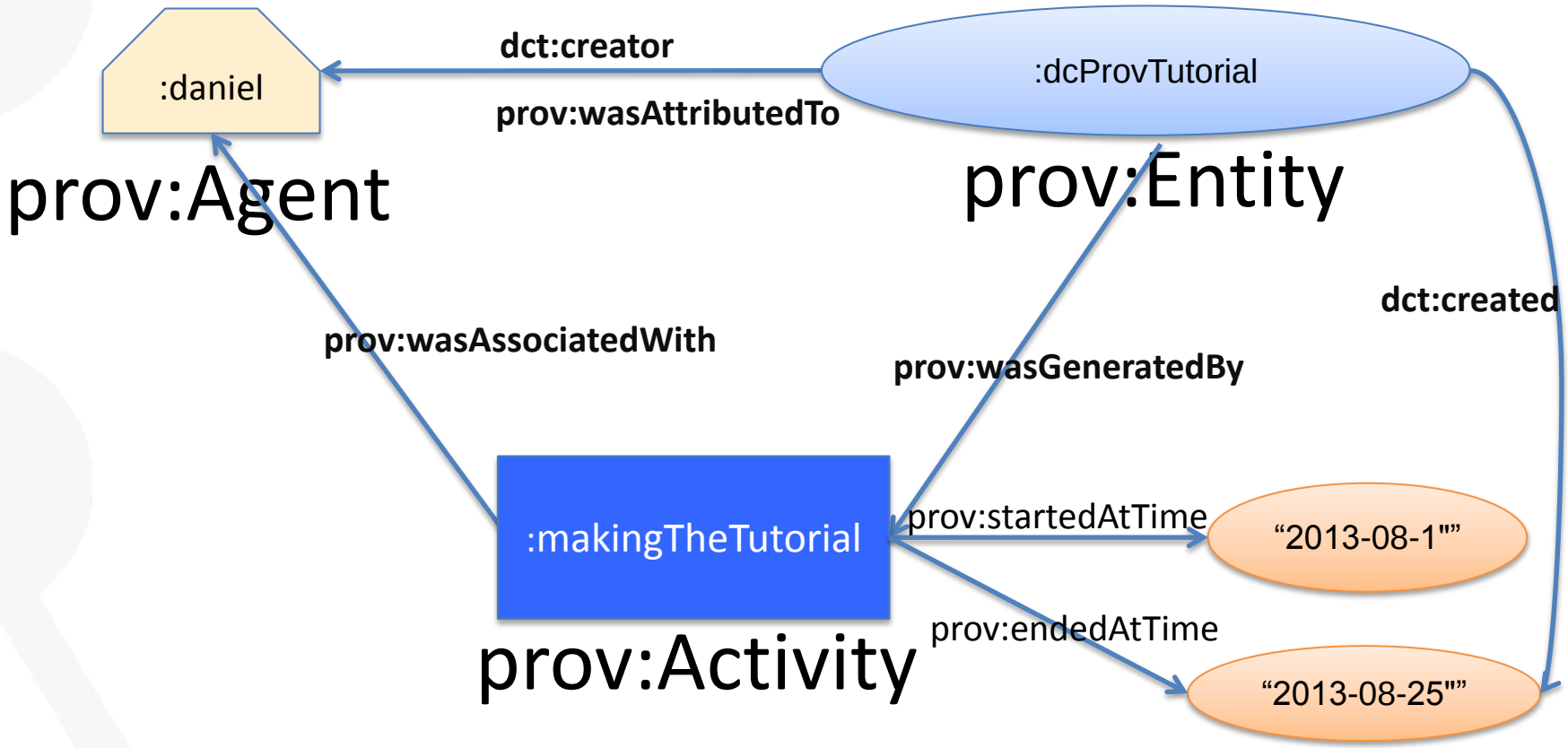


PROV-O: A very simple attribution (2)

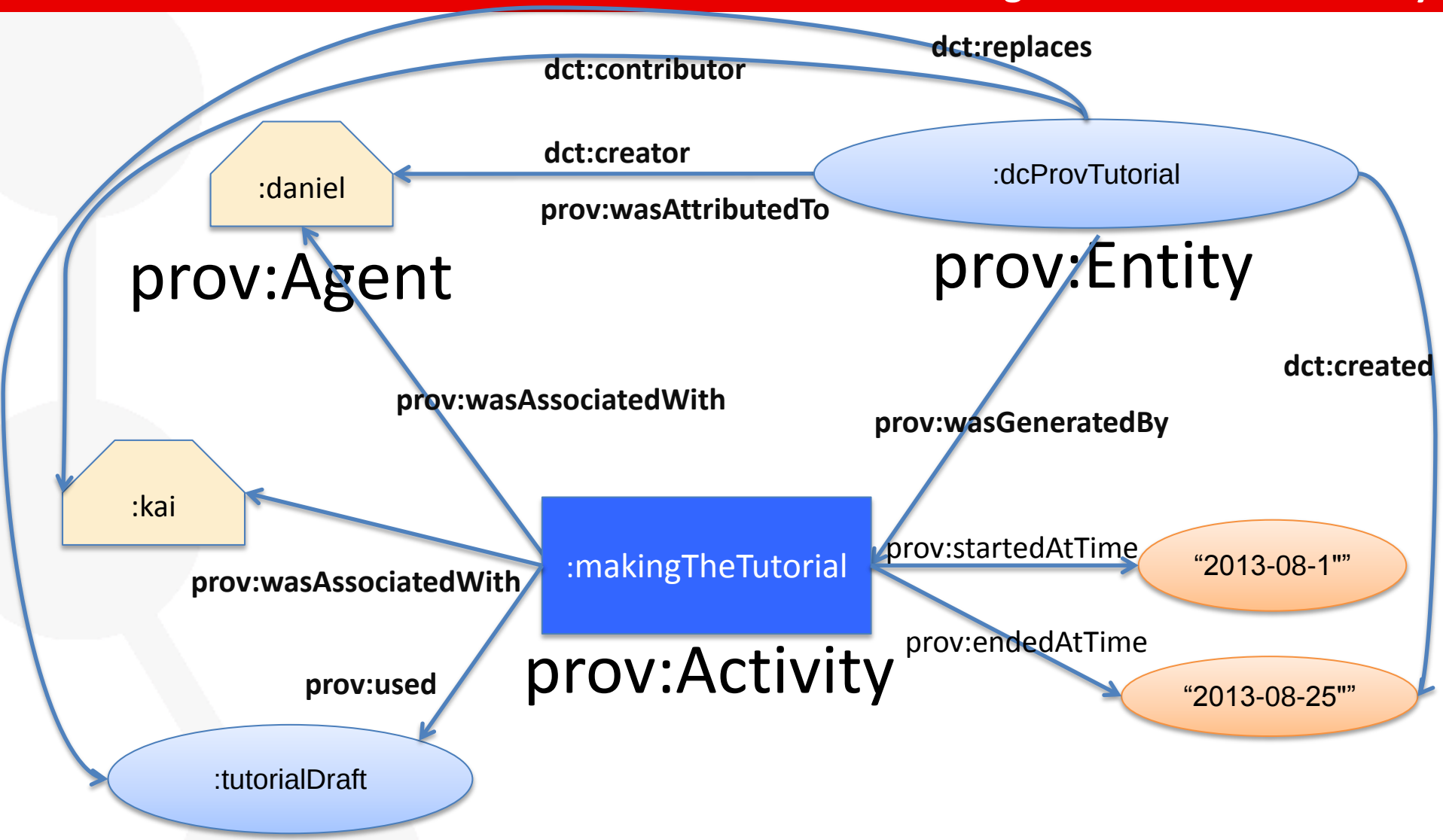



:daniel
prov:Agent
dct:creator
prov:wasAttributedTo
:dcProvTutorial
prov:Entity
prov:wasAssociatedWith
prov:wasGeneratedBy
:makingTheTutorial
prov:Activity

PROV-O: Adding metadata of the activity



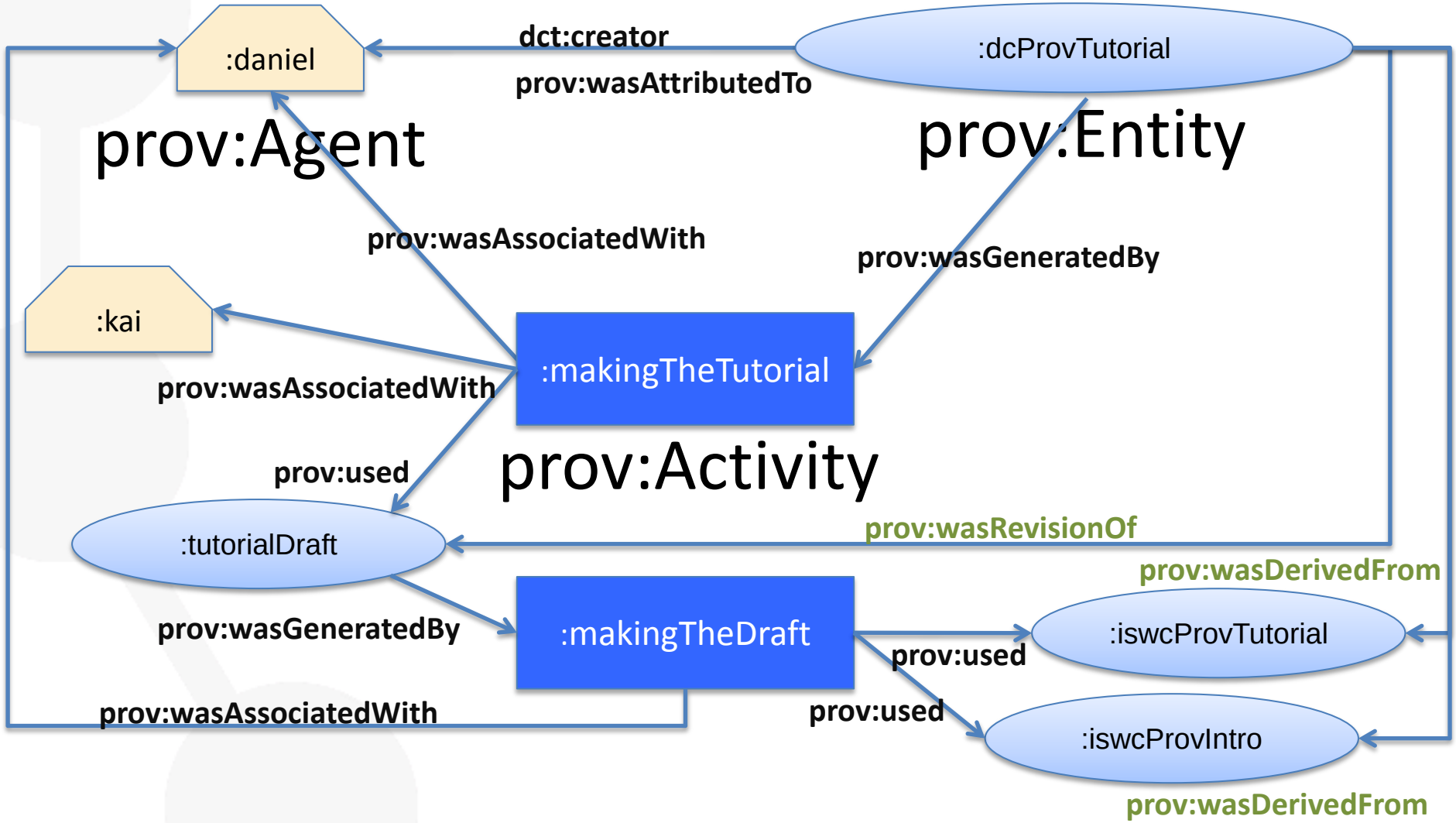
PROV-O: Adding metadata of the activity



- This example shows the fundamental notions of PROV-O
  - Entity :
    - the resources whose provenance we want to describe
  - Activity:
    - describes how entities are created and how they changed.
  - Agent:
    - responsible for the actions affecting entities.
  - Usage, generation, derivation, attribution,..
    - connections describing how entities, agents, and activities interact.

- A couple of things are still missing in the example:
  - The creation process of the draft of the tutorial.
  - The tutorial is a revision of the draft (a second version).
  - The creation process of the referenced tutorials
  - …

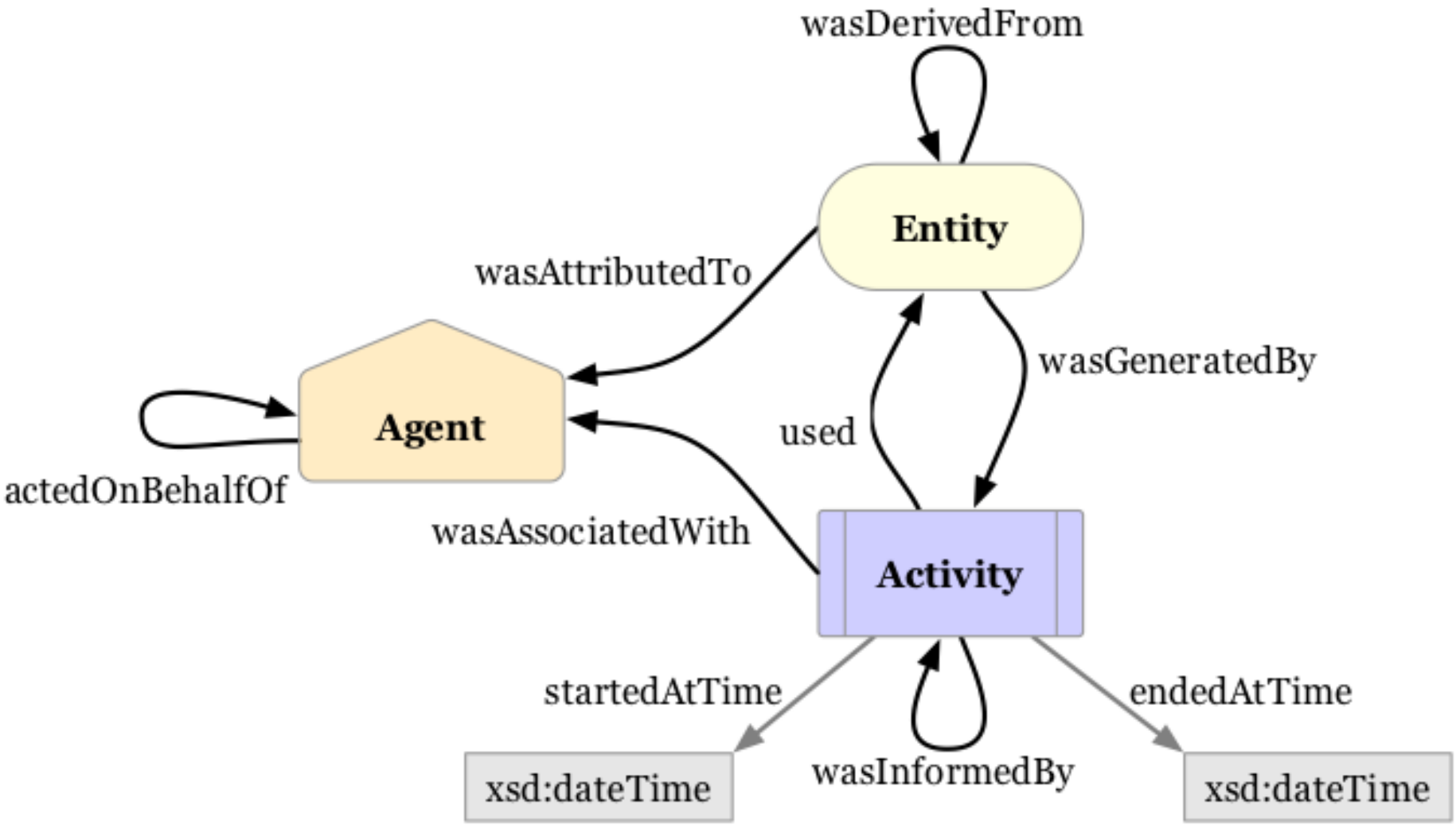
PROV-O: Making the example a little more complex (2)



PROV-O: An Overview

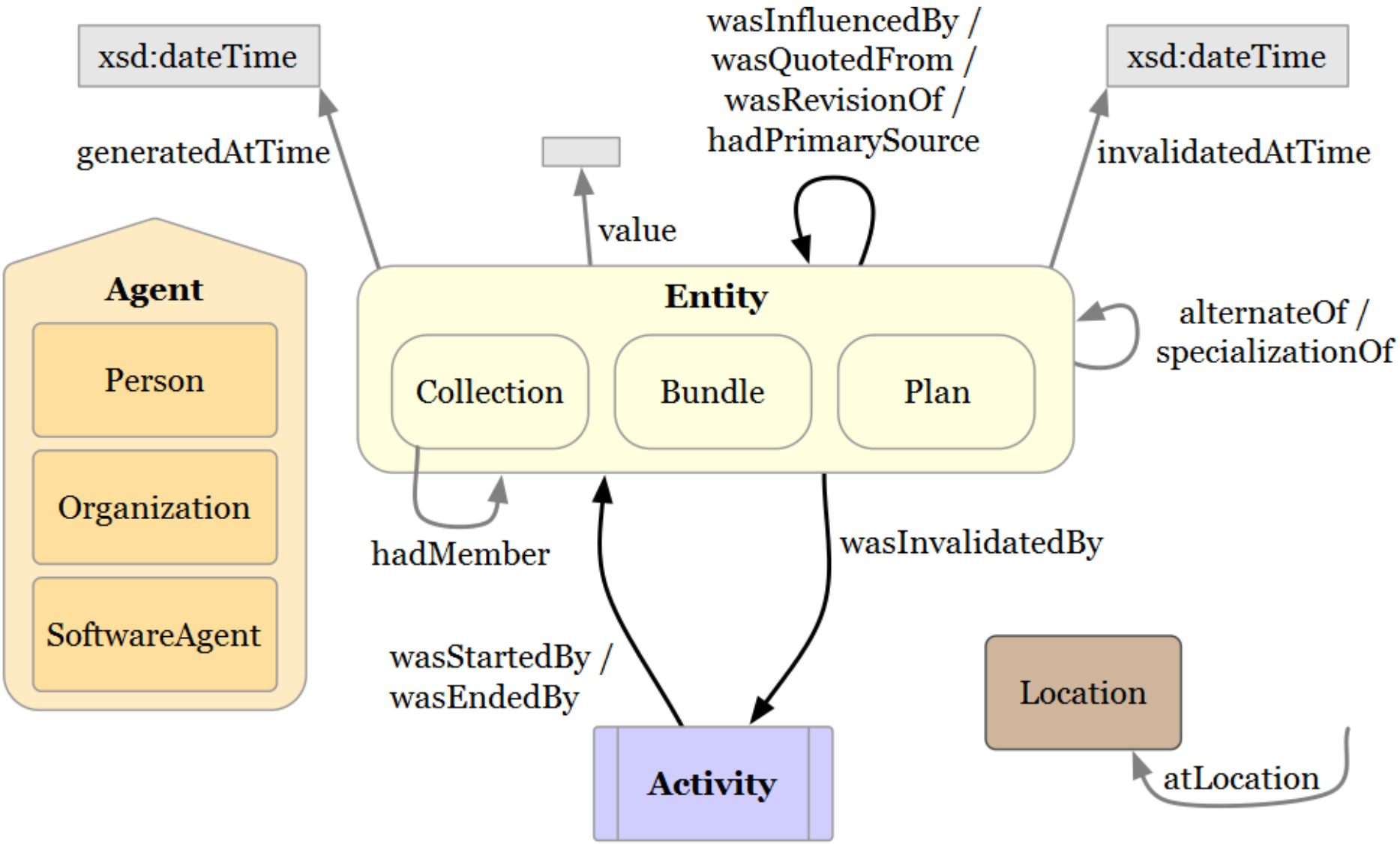
- Starting Point classes and properties: the basics.
- Expanded classes and properties: additional terms around the starting point terms for richer descriptions.
- Qualified classes and properties: for advanced provenance descriptions.

Categories of PROV terms: Starting points



- Some common specific types of agents:
  - Organization, Person, SoftwareAgent.
- Extra properties for describing versioning, influencing, invalidation, creation of entities, etc.
- Extension of the Starting points to cover generic necessities in many use cases.
  - If needed, applications may further extend this terms to their domain specific needs.

Categories of PROV terms: Starting points



- Sometimes we need to further describe the properties we have used:
  - At what time was this particular resource used?
  - What is the role of an agent in an association?
  - Where was an entity generated?
  - …
- Qualified classes provide the means for enabling such descriptions.
  - Turn relationships into classes

PROV-O: The PROV Ontology Part 1

<http://www.w3.org/ns/prov#>

A lightweight OWL ontology for interchanging provenance information

– "Simple"
– Domain neutral
– Meant to be extended
– Encodes PROV-DM's "abstract model" in RDF
– There are alternate encodings for XML, etc.

Final W3C recommendation: http://www.w3.org/TR/prov-o/

PROV in LOV:
http://lov.okfn.org/dataset/lov/details/vocabulary_prov.html

Content negotiation is enabled:

Turtle

curl -sH "Accept: text/turtle" -L http://www.w3.org/ns/prov

RDF/XML

curl -sH "Accept: application/rdf+xml" -L http://www.w3.org/ns/prov

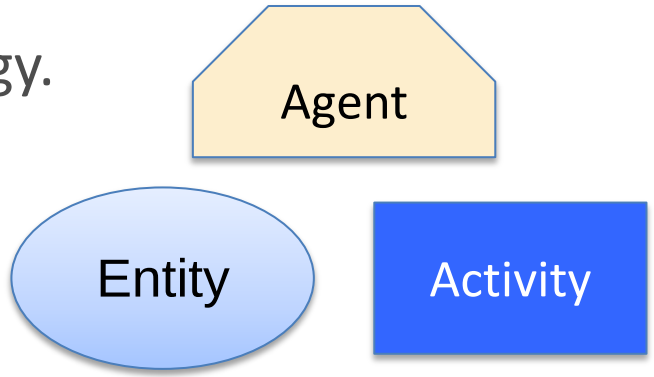
XSD

curl -sH "Accept: application/xml" -L http://www.w3.org/ns/prov

- @prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
- @prefix xsd: <http://www.w3.org/2001/XMLSchema#> .
- @prefix owl: <http://www.w3.org/2002/07/owl#> .
- @prefix prov: <http://www.w3.org/ns/prov#> .
- @prefix : <http://example.com/> .

When in doubt, prefix.cc.” (http://prefix.cc/prov)

- Starting point terms ("Simple")
  - The basics for the rest of the ontology.
  - 3 classes + 9 properties
  - Simple binary relationships

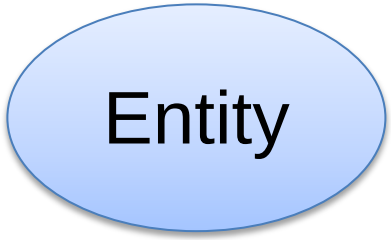


- Expanded terms ("Advanced")
  - 7 classes + 18 properties
  - Extension of the starting point terms

- Qualifying relationships ("Complex")
  - Elaborate of the 14 Starting and expanded relationships

PROV-O: Starting Points

- Entities Anything that we want to describe:
  - A document.
  - A part of a document.
  - An idea.
  - A rumor.
  - A product.
  - A contract.
  - A news article.
  - A result.
  - Etc.



"An *entity* is a physical, digital, conceptual, or other kind of thing with some fixed aspects; entities may be real or imaginary".

- Activities are any processes that used or generated entities:
  - Computing a result.
  - Making a request.
  - Writing a book.
  - Giving a presentation.
  - Creation of car.
  - Etc.

Activity

- Activities are NOT entities.

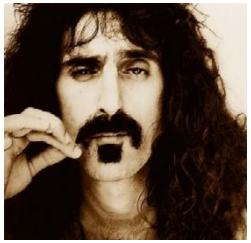
"An *activity* is something that occurs over a period of time and acts upon or with entities; it may include consuming, processing, transforming, modifying, relocating, using, or generating entities".

- Agents receive attribution for entities and are responsible for activities:
  - Creator of a document.
  - Web service accepting requests.
  - Tool or managing system.
  - An organization.
  - The student acting on behalf of the organization.
  - Etc.
- Agents can be entities.

"An *agent* is something that bears some form of responsibility for an activity taking place, for the existence of an entity, or for another agent's activity".

- Music example
  - 2 bloggers write posts about song tracks by Led Zeppelin and Frank Zappa. We know, thanks to the Music Ontology, that
    - Tracks are publications of Signals.
    - Signals are produced by recordings.
    - Recordings are made from sounds, which are realizations of a Musical Work during a musical Performance.
  - At a given point both bloggers claim their respective groups to be the authors of two songs, “A stairway to heaven”.
  - Are they referring to the same work? Which one is right?

Full example: How would you do it?



Stair way to heaven (MusicalWork)

Recording1841

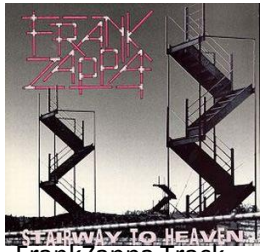
Recording999

SignalLed

SignalZappa

Towson-March1988

Performance in 1971



FrankZappa Track

SoundZappa

LedZeppelinTrack

SoundLed

Fan 1

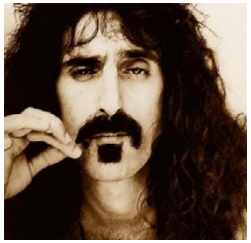
Blog post 1

Blog post 2

Fan 2

http://diymusician.cdbaby.com/wp-content/uploads/2013/06/Millenial-282x300.jpg

Full example: How would you do it?



Entity
Stair way to heaven (MusicalWork)

Recording1841

Recording999

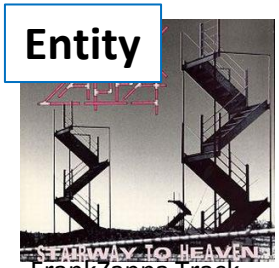
Entity
SignalLed

Entity
SignalZappa

Towson-March1988

Performance in 1971

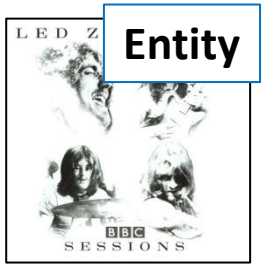
Entity



FrankZappa Track

Entity
SoundZappa

Entity



LedZeppelinTrack

Entity
SoundLed

Fan 1

Entity

Blog post 1

Entity

Blog post 2

Fan 2

http://diymusician.cdbaby.com/wp-content/uploads/2013/06/Millenial-282x300.jpg

Full example: How would you do it?

Entity
Stair way to heaven (MusicalWork)

Recording1841

Recording999

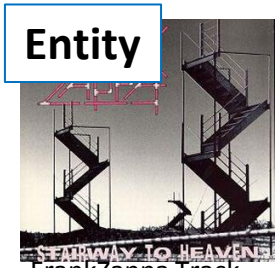
Entity
SignalLed

Entity
SignalZappa

Towson-March1988

Performance in 1971

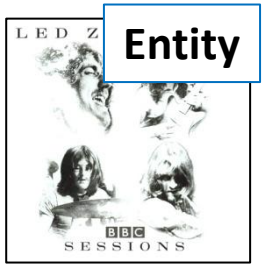
Entity



FrankZappa Track

Entity
SoundZappa

Entity



LedZeppelinTrack

Entity
SoundLed

Fan 1

Entity

Blog post 1

Entity

Blog post 2

Fan 2

http://diymusician.cdbaby.com/wp-content/uploads/2013/06/Millenial-282x300.jpg

Full example: How would you do it?

Agent

Entity

Stair way to heaven (MusicalWork)

Activity

Recordin

Agent

Activity

Recordin

Entity

SignalLed

Entity

SignalZappa

Activity

Towso

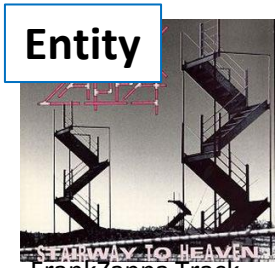
March1988

Activity

Perfor

in 1971

Entity

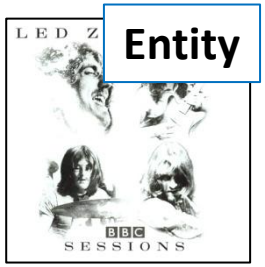


FrankZappa Track

Entity

SoundZappa

Entity



LedZeppelinTrack

Entity

SoundLed

Agent

Fan 1

Entity

Blog post 1

Entity

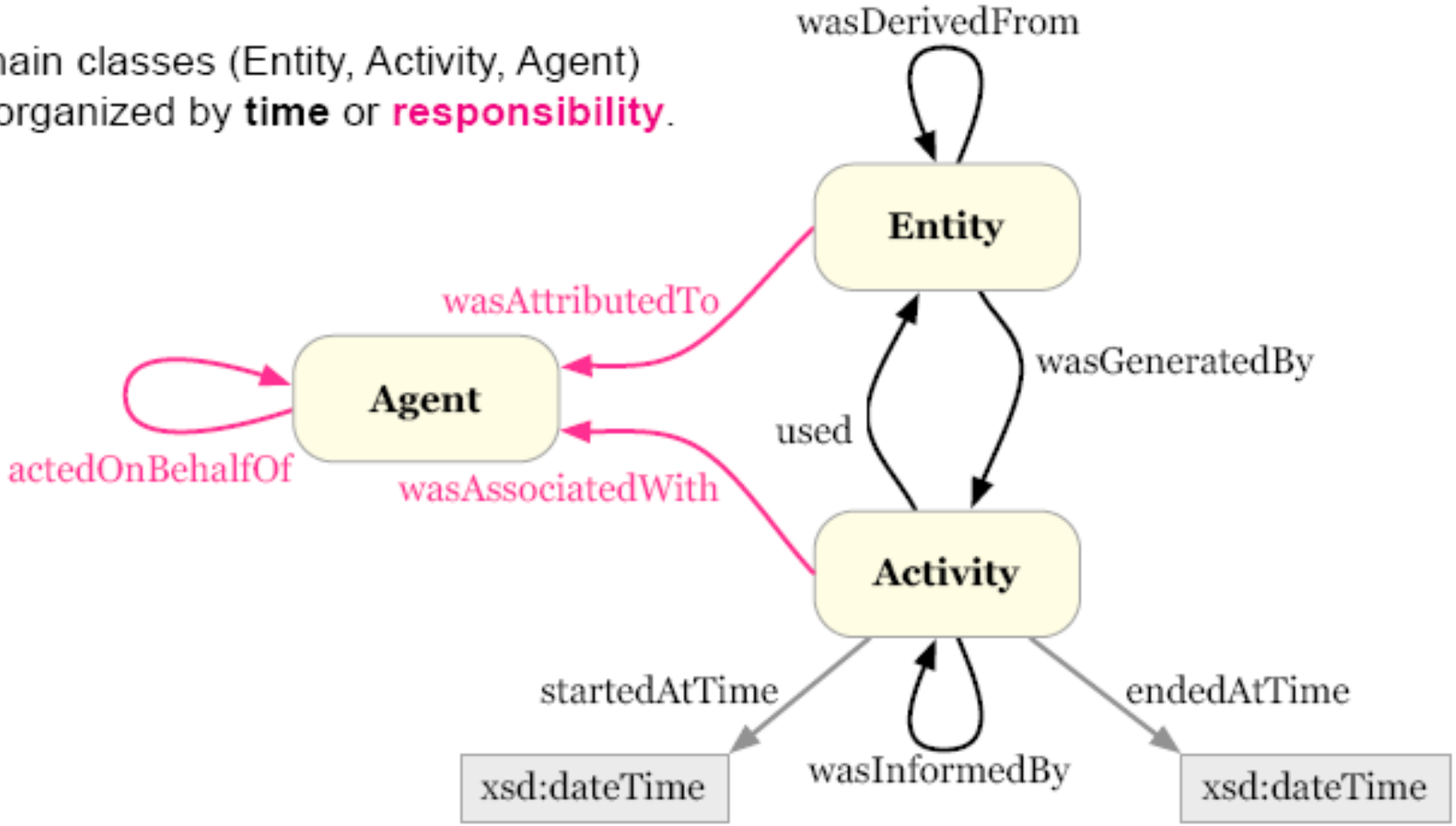
Blog post 2

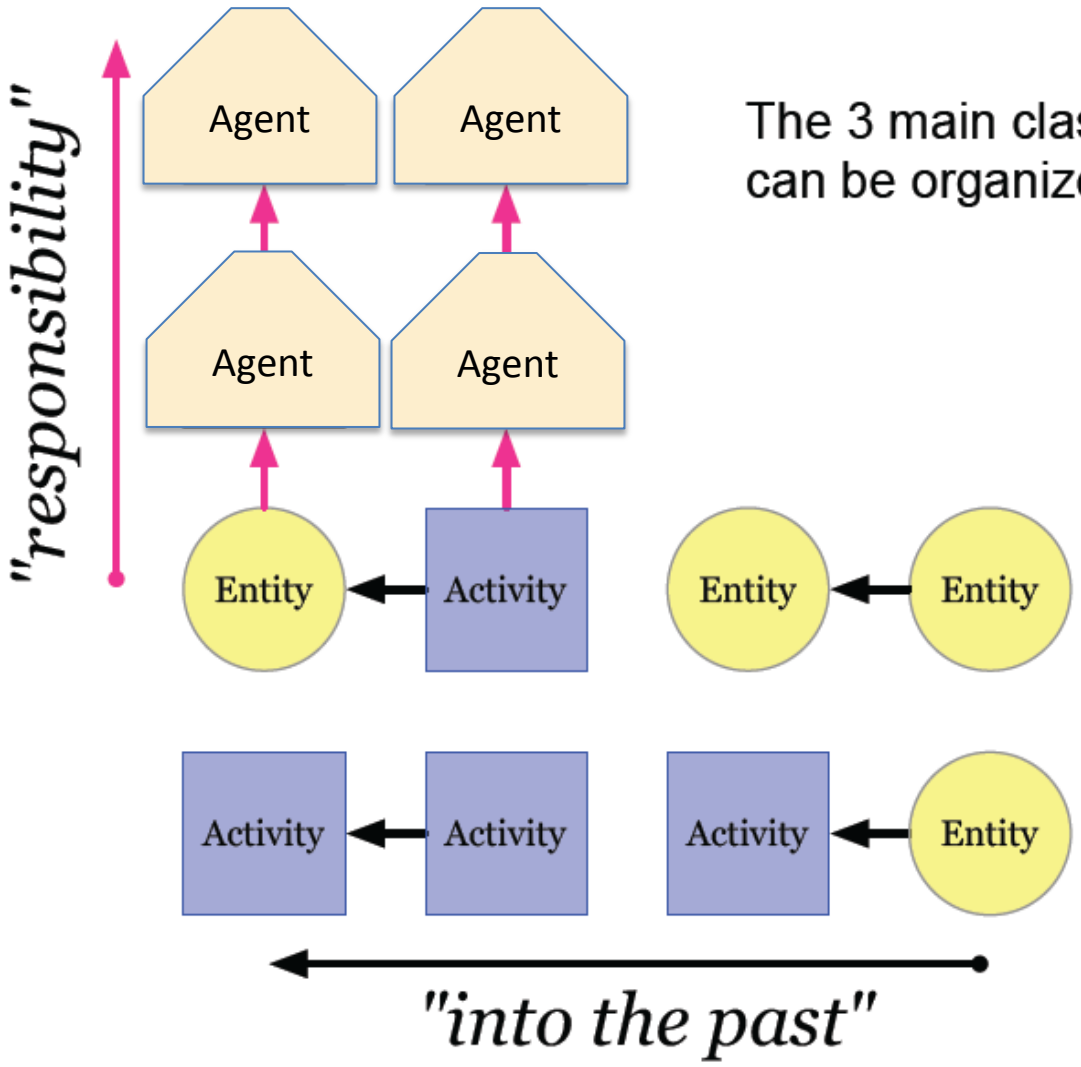
Agent

Fan 2

http://diymusician.cdbaby.com/wp-content/uploads/2013/06/Millenial-282x300.jpg

The 3 main classes (Entity, Activity, Agent) can be organized by **time** or **responsibility**.



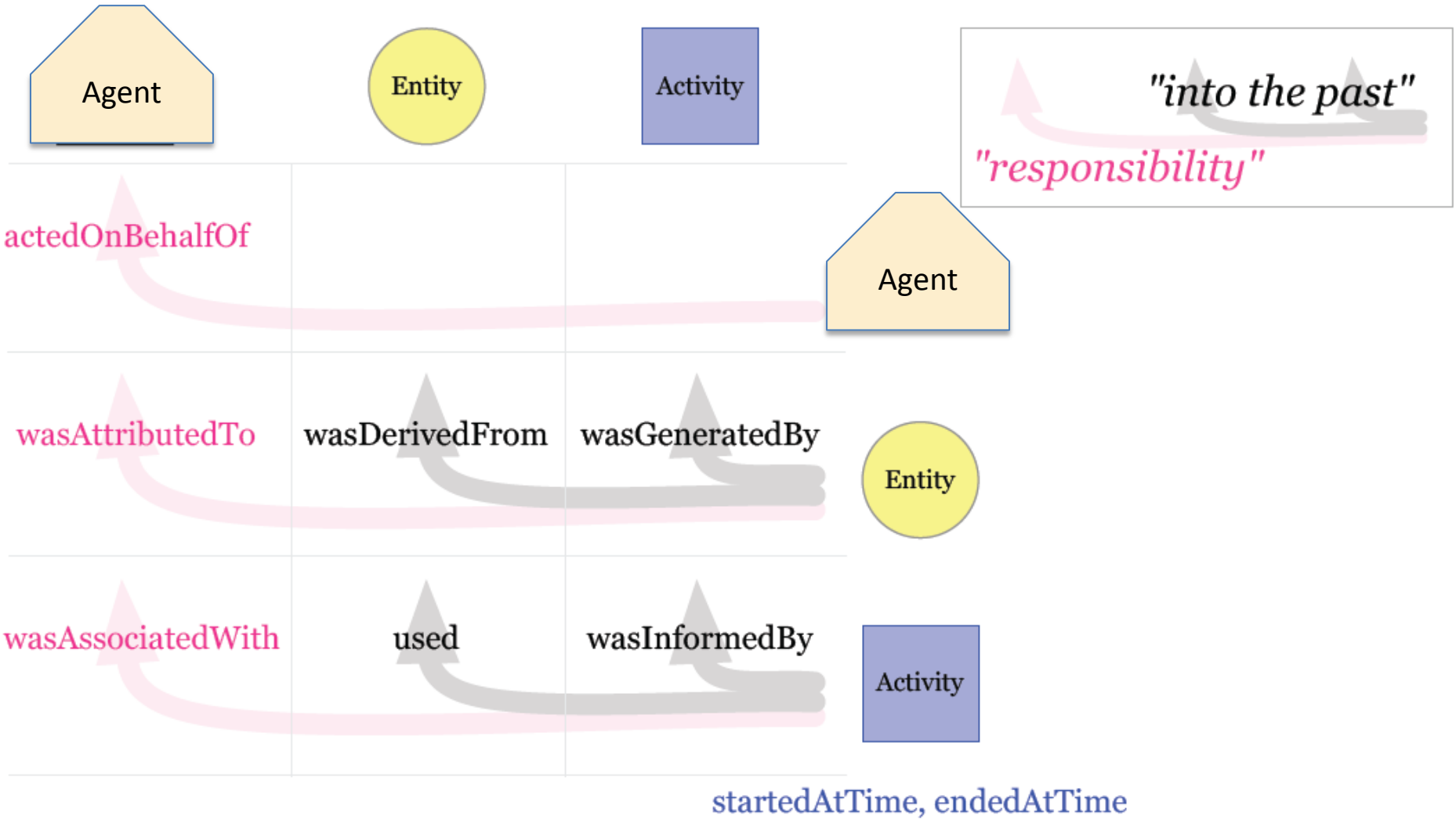


The 3 main classes (Entity, Activity, Agent) can be organized by **time** or **responsibility**.

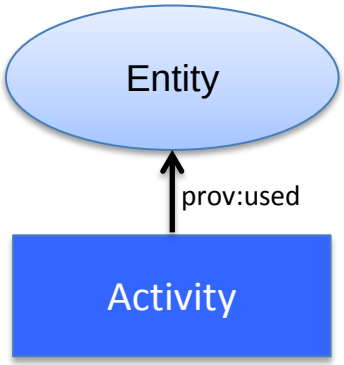
- actedOnBehalfOf
- wasAttributedTo
- wasAssociatedWith

- used, wasDerivedFrom
- wasInformedBy, wasGeneratedBy

- startedAtTime, endedAtTime


Agent
Entity
Activity
"into the past"
"responsibility"
actedOnBehalfOf
Agent
wasAttributedTo
wasDerivedFrom
wasGeneratedBy
Entity
wasAssociatedWith
used
wasInformedBy
Activity
startedAtTime, endedAtTime

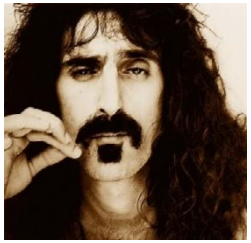
- Usage is crucial for describing the entities which participated in an activity:
  - The references used for creating a document.
  - The query used to obtain a result.
  - The inputs of a computational process.
  - Etc.



"*Usage* is the beginning of utilizing an entity by an activity. Before usage, the activity had not begun to utilize this entity and could not have been affected by the entity".

- Music example
  - 2 bloggers write posts about song tracks by Led Zeppelin and Frank Zappa. We know, thanks to the Music Ontology, that
    - Tracks are publications of Signals.
    - Signals are produced by recordings.
    - **Recordings are made from sounds, which are realizations of a Musical Work during a musical Performance**.

Full example: Usage



Stair way to heaven (MusicalWork)

Recording1841

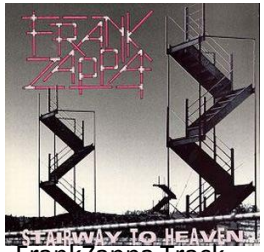
Recording999

SignalLed

SignalZappa

Towson-March1988

Performance in 1971



FrankZappa Track

SoundZappa

LedZeppelinTrack

SoundLed

Fan 1

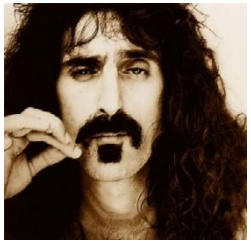
Blog post 1

Blog post 2

Fan 2

http://diymusician.cdbaby.com/wp-content/uploads/2013/06/Millenial-282x300.jpg

Full example: Usage



Stair way to heaven (MusicalWork)

Recording1841

prov:used

prov:used

Recording999

SignalLed

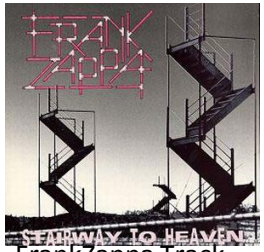
prov:used

prov:used

Towson-March1988

Performance in 1971

SignalZappa



FrankZappa Track

SoundZappa

LedZeppelinTrack

SoundLed

Fan 1

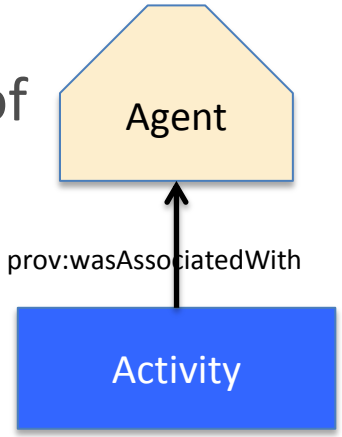
Blog post 1

Blog post 2

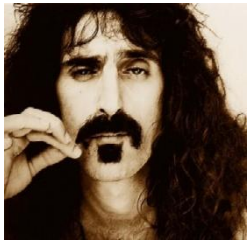
Fan 2

http://diymusician.cdbaby.com/wp-content/uploads/2013/06/Millenial-282x300.jpg

- Association is crucial for assigning responsibility:
  - Who is the responsible for a document?
  - Who is the responsible for the creation of a result of a computational experiment?
  - Who is responsible for the development of a product/contract, etc?
  - Etc.



"An activity *association* is an assignment of responsibility to an agent for an activity, indicating that the agent had a role in the activity. It further allows for a plan to be specified, which is the plan intended by the agent to achieve some goals in the context of this activity".



Stair way to heaven (MusicalWork)

Recording1841

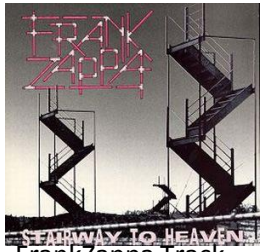
Recording999

SignalLed

SignalZappa

Towson-March1988

Performance in 1971



FrankZappa Track

SoundZappa

LedZeppelinTrack

SoundLed

Fan 1

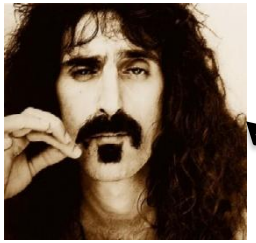
Blog post 1

Blog post 2

Fan 2

http://diymusician.cdbaby.com/wp-content/uploads/2013/06/Millenial-282x300.jpg

Full example: Association



Stair way to heaven (MusicalWork)

Recording1841

Recording999

SignalLed

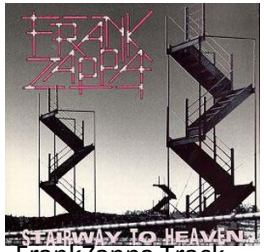
prov:wasAssociatedWith

prov:wasAssociatedWith

Towson-March1988

Performance in 1971

SignalZappa



FrankZappa Track

SoundZappa

LedZeppelinTrack

SoundLed

Fan 1

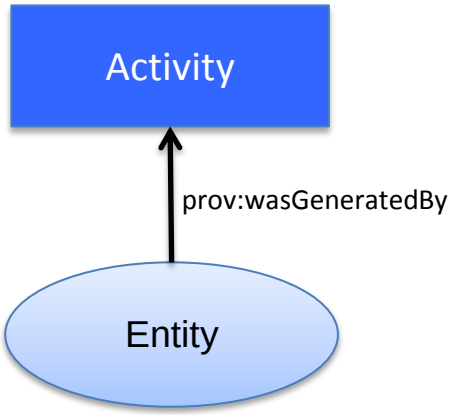
Blog post 1

Blog post 2

Fan 2

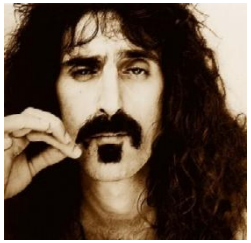
http://diymusician.cdbaby.com/wp-content/uploads/2013/06/Millenial-282x300.jpg

- Generation is crucial for describing entities and their origin:
  - How was a document generated?.
  - How is a computational result created?.
  - How has an entity been modified?.
  - How was a result validated?
  - Etc.



"*Generation* is the completion of production of a new entity by an activity. This entity did not exist before generation and becomes available for usage after this generation".

- Music example
  - 2 bloggers write posts about song tracks by Led Zeppelin and Frank Zappa. We know, thanks to the Music Ontology, that
    - Tracks are publications of Signals.
    - **Signals are produced by recordings.**
    - Recordings are made from sounds, which are realizations of a Musical Work during a musical Performance.



Stair way to heaven (MusicalWork)

Recording1841

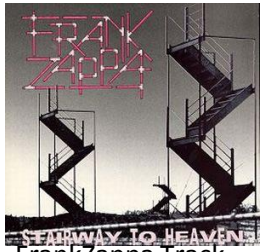
Recording999

SignalLed

SignalZappa

Towson-March1988

Performance in 1971



FrankZappa Track

SoundZappa

LedZeppelinTrack

SoundLed

Fan 1

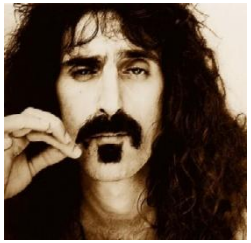
Blog post 1

Blog post 2

Fan 2

http://diymusician.cdbaby.com/wp-content/uploads/2013/06/Millenial-282x300.jpg

Full example: Generation



Stair way to heaven (MusicalWork)

Recording1841

prov:wasGeneratedBy

Recording999

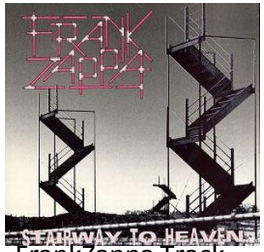
prov:wasGeneratedBy

SignalLed

SignalZappa

Towson-March1988

Performance in 1971



FrankZappa Track

SoundZappa

LedZeppelinTrack

SoundLed

Fan 1

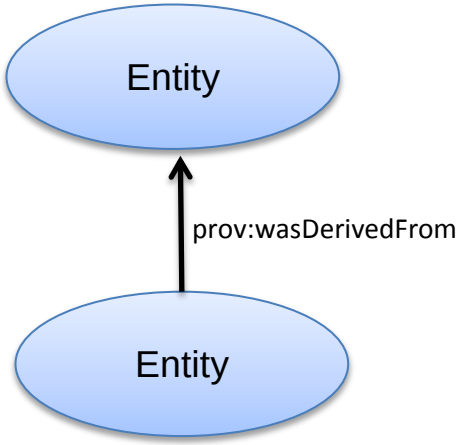
Blog post 1

Blog post 2

Fan 2

http://diymusician.cdbaby.com/wp-content/uploads/2013/06/Millenial-282x300.jpg

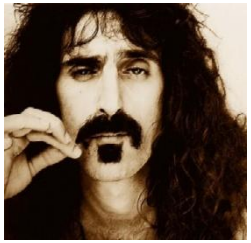
- Derivation is used for describing dependencies among entities:
  - Are the contents of a document based on other entities?.
  - How does a computational result depend on external databases?.
  - Which resources have influenced at some point this entity?.
  - Etc.



"A *Derivation* is a transformation of an entity into another, an update of an entity resulting in a new one, or the construction of a new entity based on a pre-existing entity".

- Music example
  - 2 bloggers write posts about song tracks by Led Zeppelin and Frank Zappa. We know, thanks to the Music Ontology, that
    - **Tracks are publications of Signals**.
    - Signals are produced by recordings**.**
    - Recordings are made from sounds, which are realizations of a Musical Work during a musical Performance.

Full example: Derivation



Stair way to heaven (MusicalWork)

Recording1841

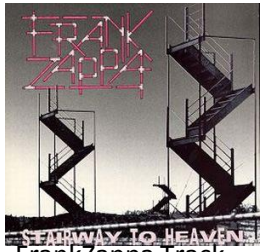
Recording999

SignalLed

SignalZappa

Towson-March1988

Performance in 1971



FrankZappa Track

SoundZappa

LedZeppelinTrack

SoundLed

Fan 1

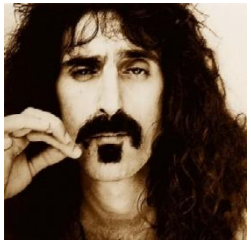
Blog post 1

Blog post 2

Fan 2

http://diymusician.cdbaby.com/wp-content/uploads/2013/06/Millenial-282x300.jpg

Full example: Derivation



Stair way to heaven (MusicalWork)

Recording1841

Recording999

SignalLed

Towson-March1988

Performance in 1971

SignalZappa

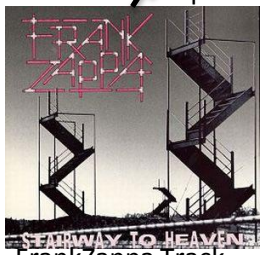
prov:wasDerivedFrom

prov:wasDerivedFrom

LedZeppelinTrack

SoundZappa

SoundLed



FrankZappa Track

Fan 1

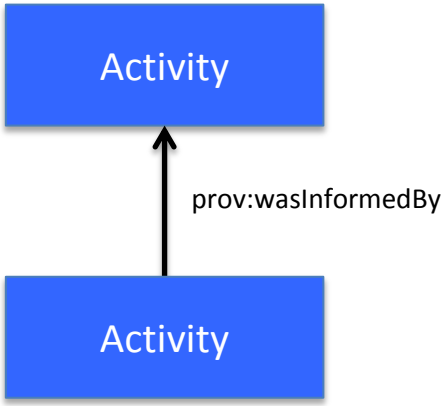
Blog post 1

Blog post 2

Fan 2

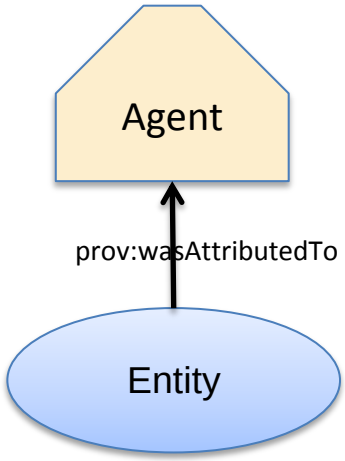
http://diymusician.cdbaby.com/wp-content/uploads/2013/06/Millenial-282x300.jpg

- Communication is used for describing dependencies between activities:
  - Which activities precede the current one?.
  - What are the steps required for executing the current query?.
  - Etc.



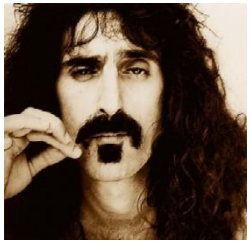
"*Communication* (wasInformedBy) is the exchange of some unspecified entity by two activities, one activity using some entity generated by the other".

- Attribution is key for giving credit to someone:
  - Who is the author of a particular document?.
  - Which tool/Software has been used to generate a result?.
  - Who has created a this dataset?
  - Etc.



"*Attribution* is the ascribing of an entity to an agent".

Full example: Attribution



Stair way to heaven (MusicalWork)

Recording1841

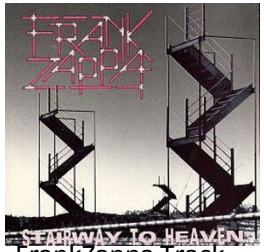
Recording999

SignalLed

SignalZappa

Towson-March1988

Performance in 1971



FrankZappa Track

SoundZappa

LedZeppelinTrack

SoundLed

prov:wasAttributedTo

Fan 1

Blog post 1

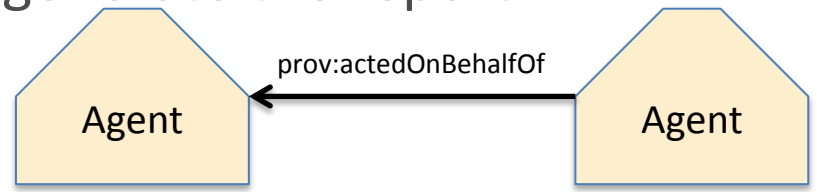
prov:wasAttributedTo

Blog post 2

Fan 2

http://diymusician.cdbaby.com/wp-content/uploads/2013/06/Millenial-282x300.jpg

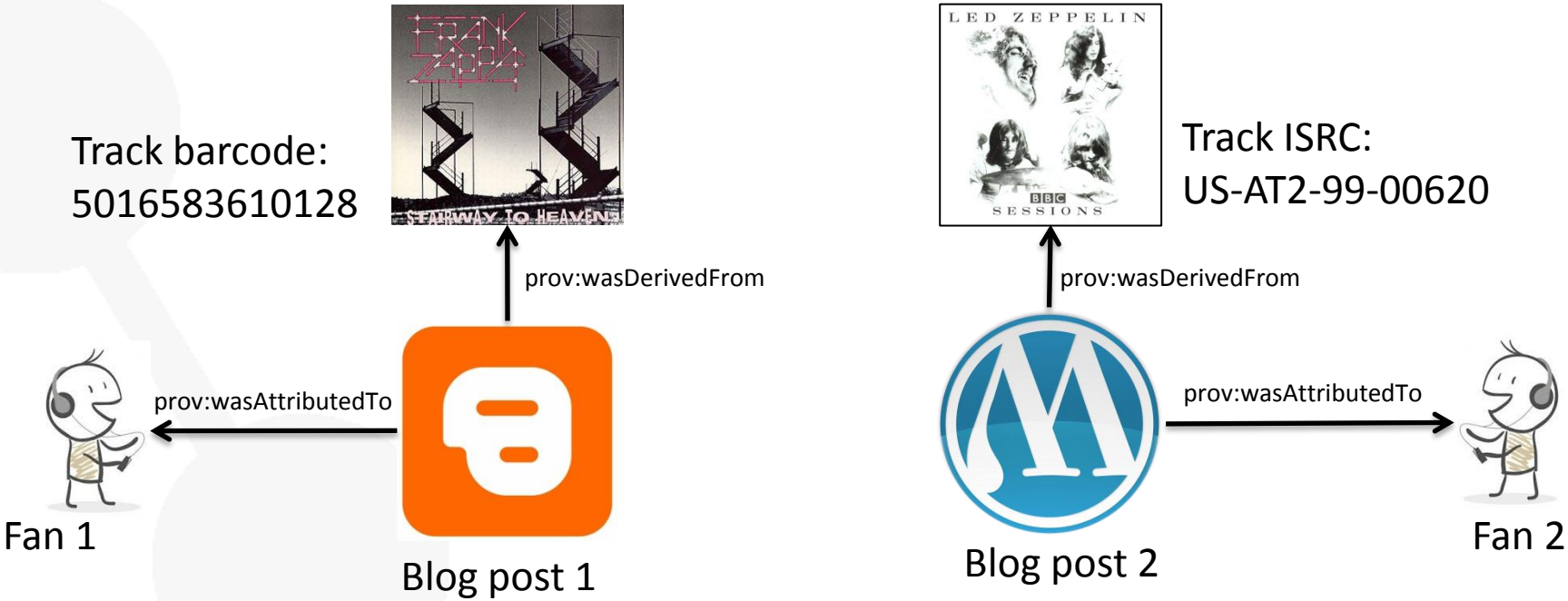
- Delegation is used to specify responsibility between agents:
  - Who is the responsible for the generation of the result of a computational experiment (which acted on behalf of UPM?).
  - Which user activated the tool to generate the report?.
  - Etc.



"*Delegation (*actedOnBehalfOf*)* is the assignment of authority and responsibility to an agent (by itself or by another agent) to carry out a specific activity as a delegate or representative, while the agent it acts on behalf of retains some responsibility for the outcome of the delegated work".

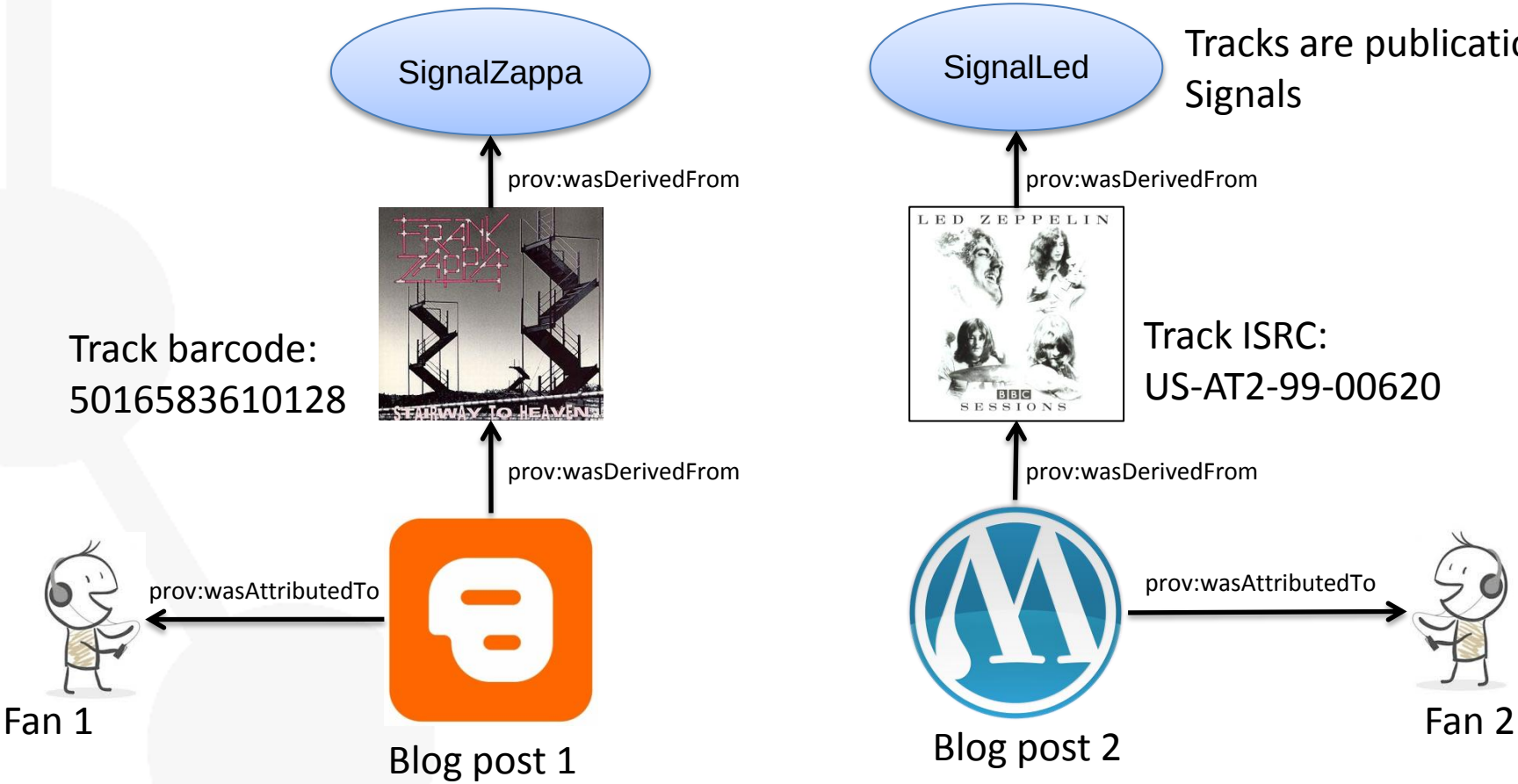
http://diymusician.cdbaby.com/wp-content/uploads/2013/06/Millenial-282x300.jpg

Full example



http://diymusician.cdbaby.com/wp-content/uploads/2013/06/Millenial-282x300.jpg

Full example



http://diymusician.cdbaby.com/wp-content/uploads/2013/06/Millenial-282x300.jpg

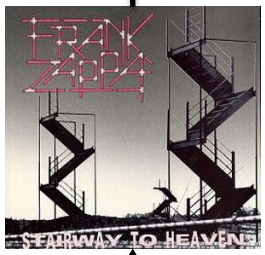
Information from recording not available

Recording999

prov:wasGeneratedBy

SignalZappa

prov:wasDerivedFrom



Track barcode: 5016583610128

prov:wasDerivedFrom

prov:wasAttributedTo

Fan 1

Blog post 1

Recording1841

Recording 1841 in Playhouse Theatre, Northumberland Av

prov:wasGeneratedBy

SignalLed

Tracks are publications of Signals

prov:wasDerivedFrom

Track ISRC: US-AT2-99-00620

prov:wasDerivedFrom

prov:wasAttributedTo

Blog post 2

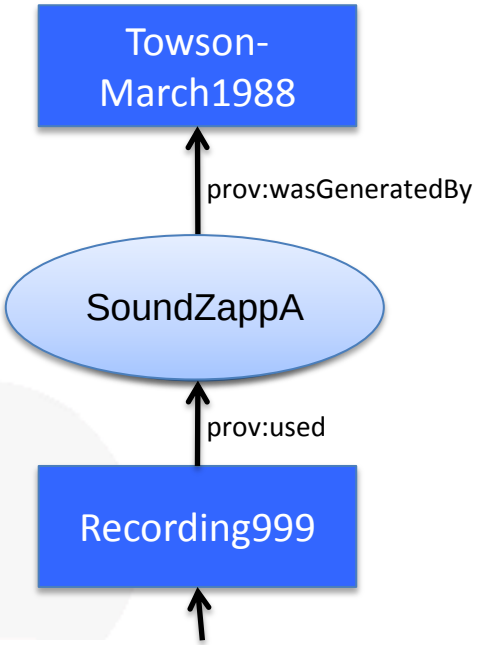
Fan 2

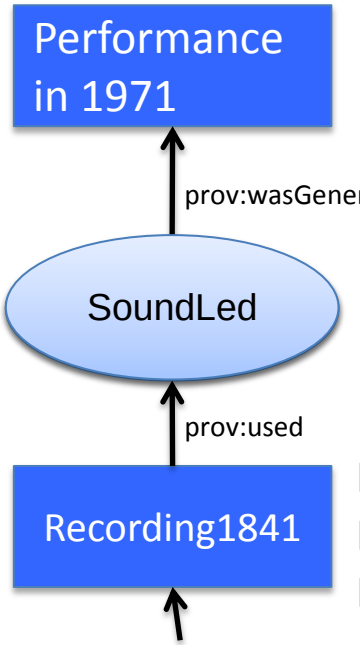
http://diymusician.cdbaby.com/wp-content/uploads/2013/06/Millenial-282x300.jpg

Information from recording not available

Realisation of a MusicalWork during a musical Performance

Recording 1841 in Playhouse Theatre, Northumberland Av

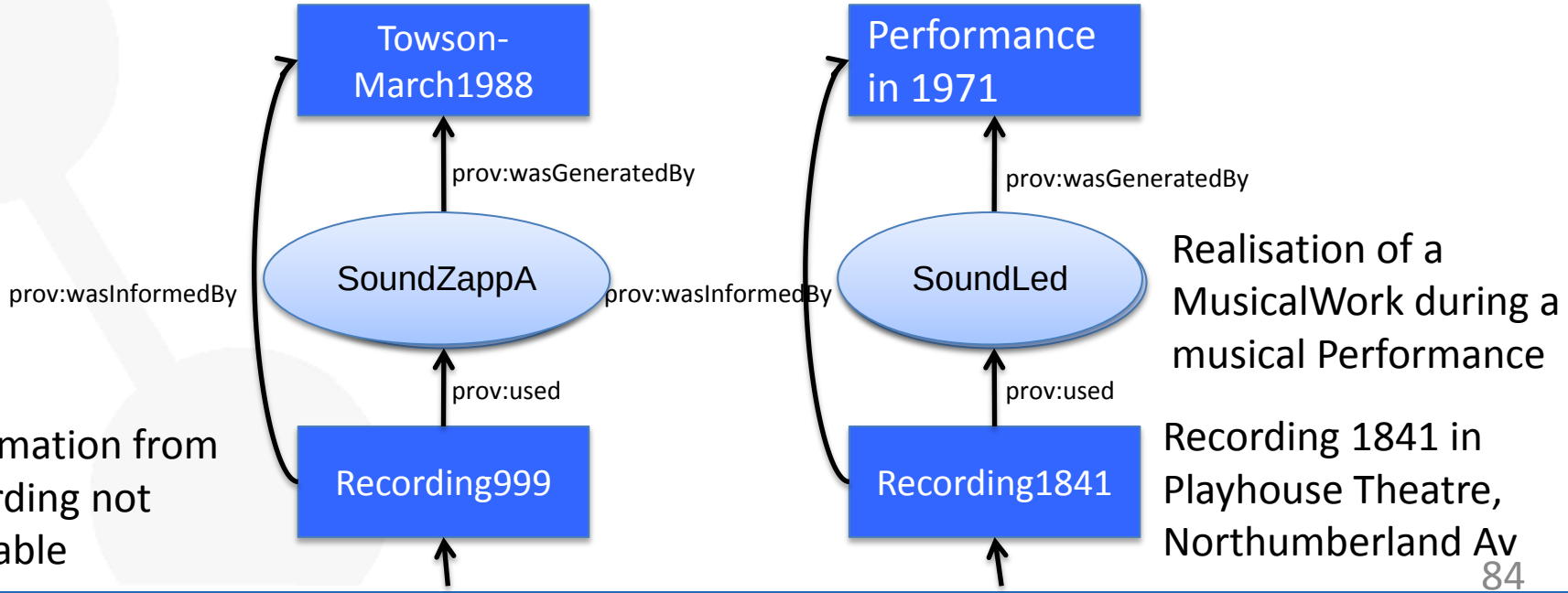




Information from recording not available

Realisation of a MusicalWork during a musical Performance

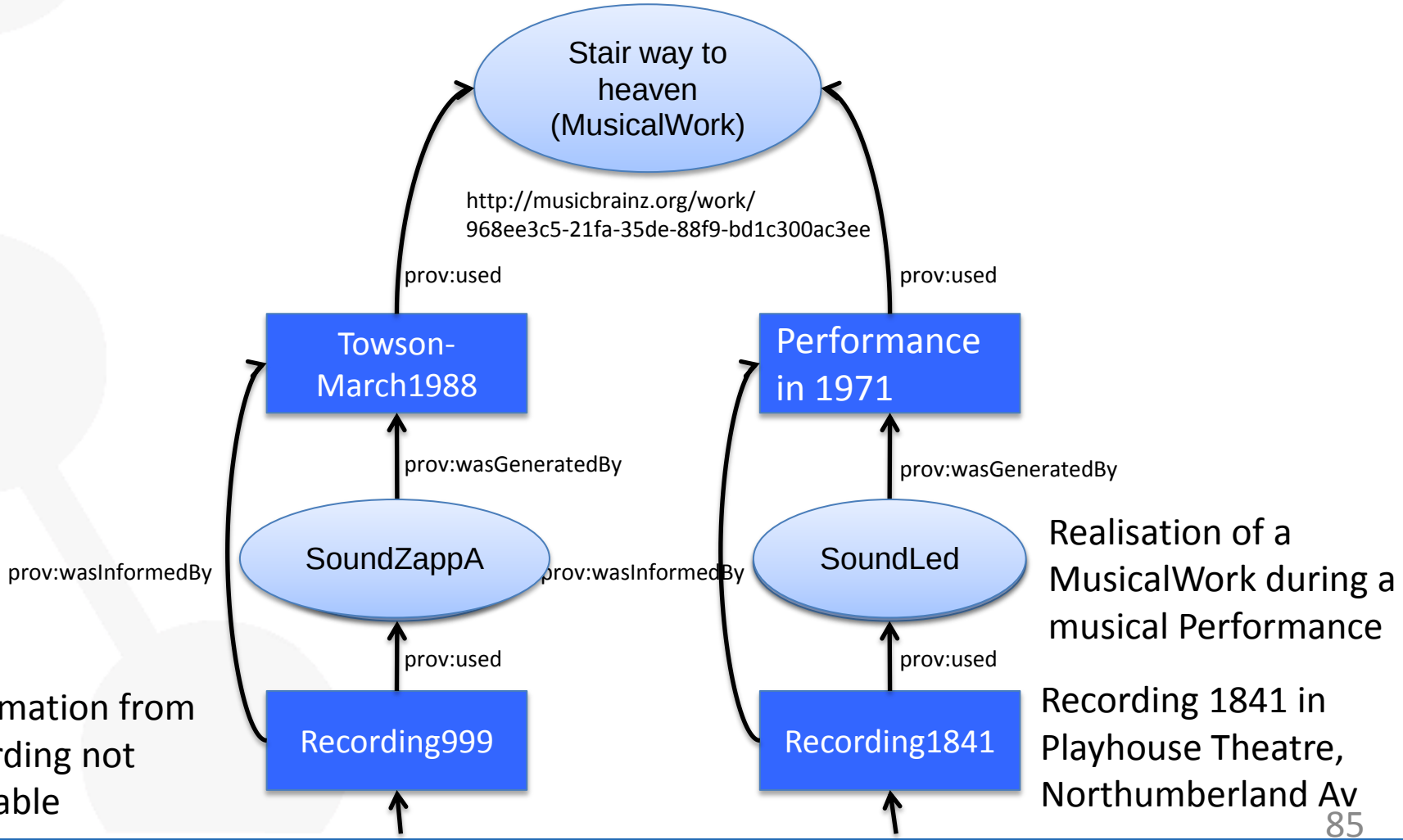
Recording 1841 in Playhouse Theatre, Northumberland Av



Information from recording not available

Realisation of a MusicalWork during a musical Performance

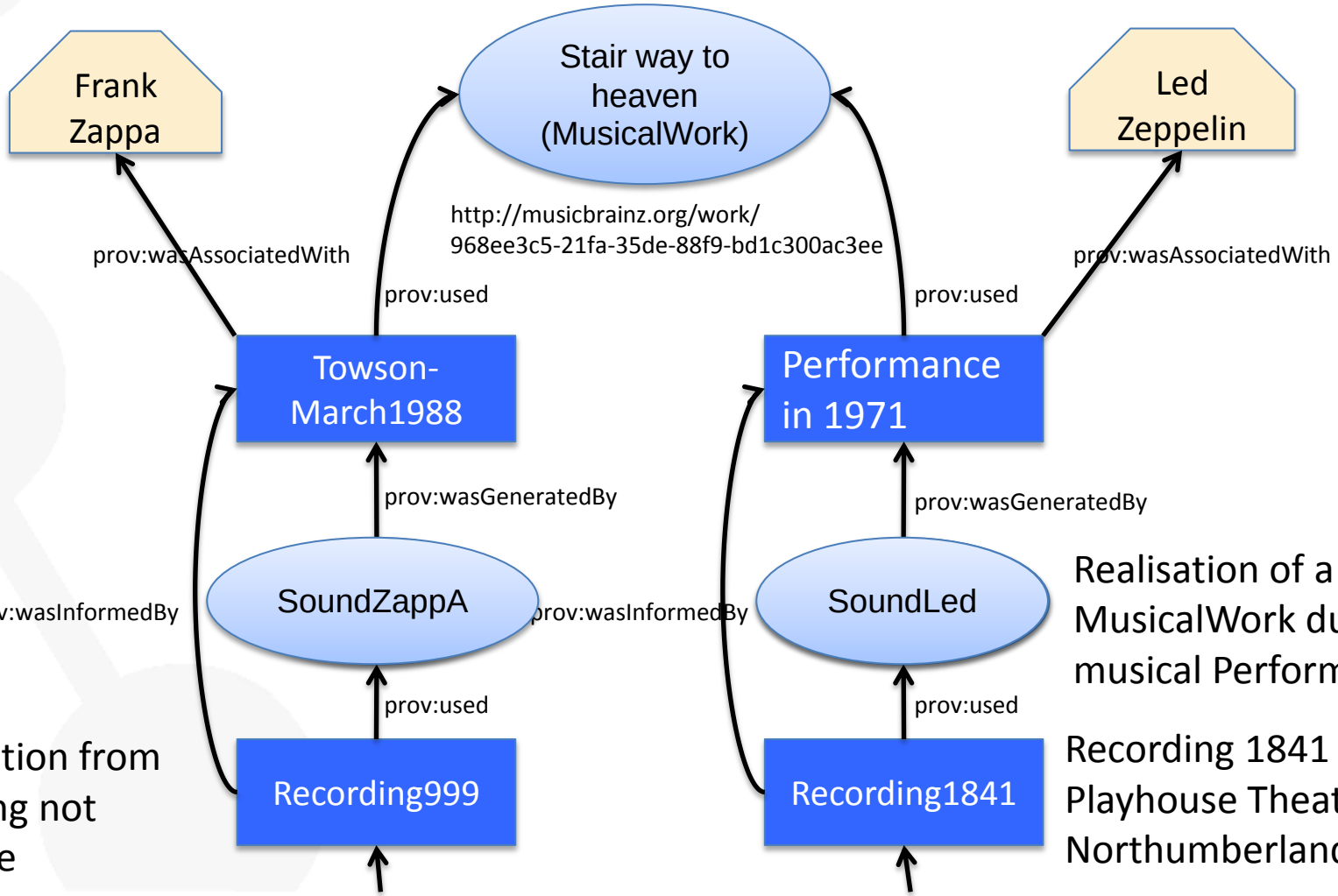
Recording 1841 in Playhouse Theatre, Northumberland Av



Realisation of a MusicalWork during a musical Performance

Information from recording not available

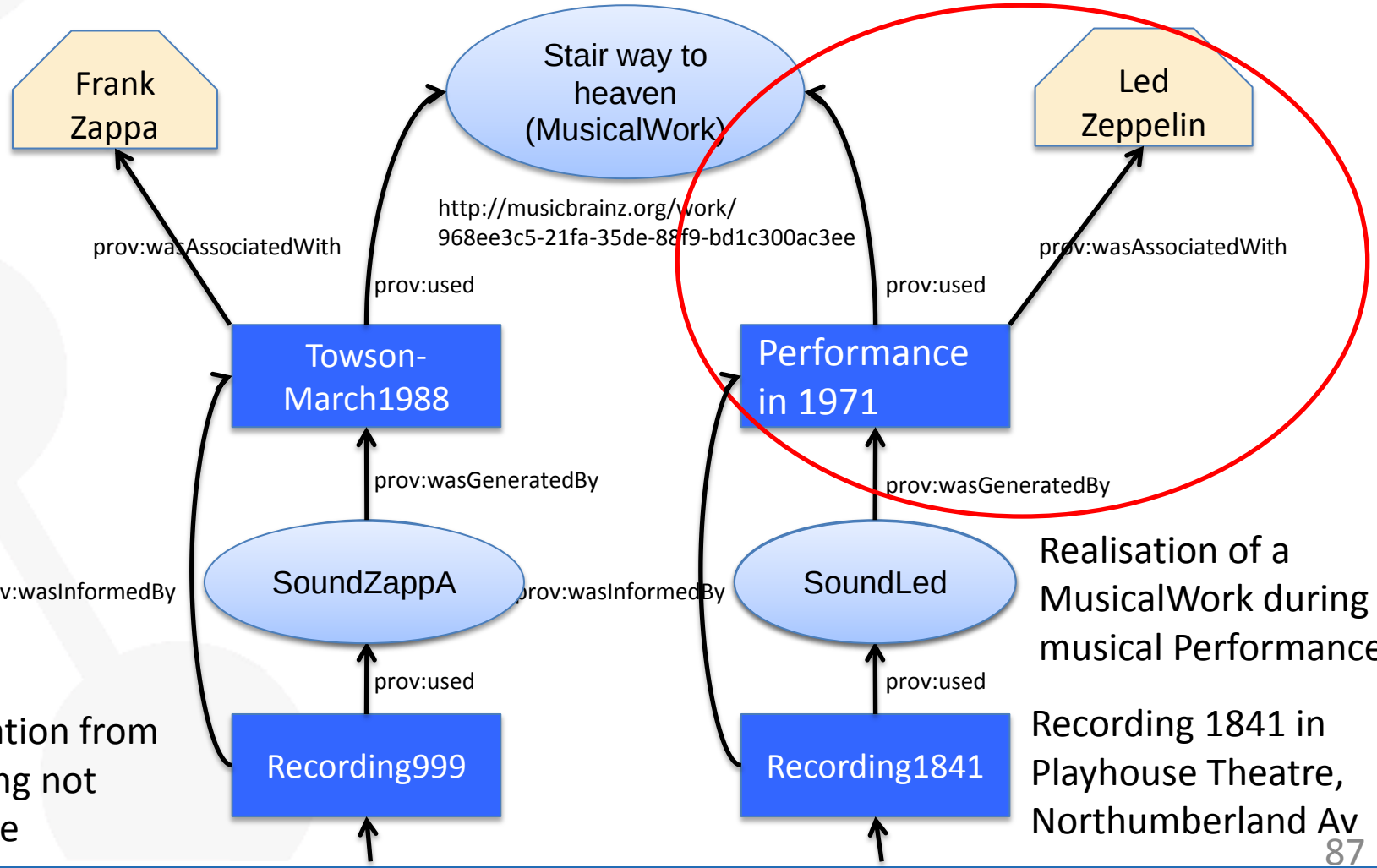
Recording 1841 in Playhouse Theatre, Northumberland Av



Realisation of a MusicalWork during a musical Performance

Information from recording not available

Recording 1841 in Playhouse Theatre, Northumberland Av


Frank Zappa
Stair way to heaven (MusicalWork)
Led Zeppelin
http://musicbrainz.org/work/968ee3c5-21fa-35de-88f9-bd1c300ac3ee
prov:wasAssociatedWith
prov:used
prov:used
prov:wasAssociatedWith
Towson-March1988
Performance in 1971
prov:wasGeneratedBy
prov:wasGeneratedBy
prov:wasInformedBy
SoundZappA
prov:wasInformedBy
SoundLed
prov:used
prov:used
Recording999
Recording1841
Realisation of a MusicalWork during a musical Performance
Information from recording not available
Recording 1841 in Playhouse Theatre, Northumberland Av

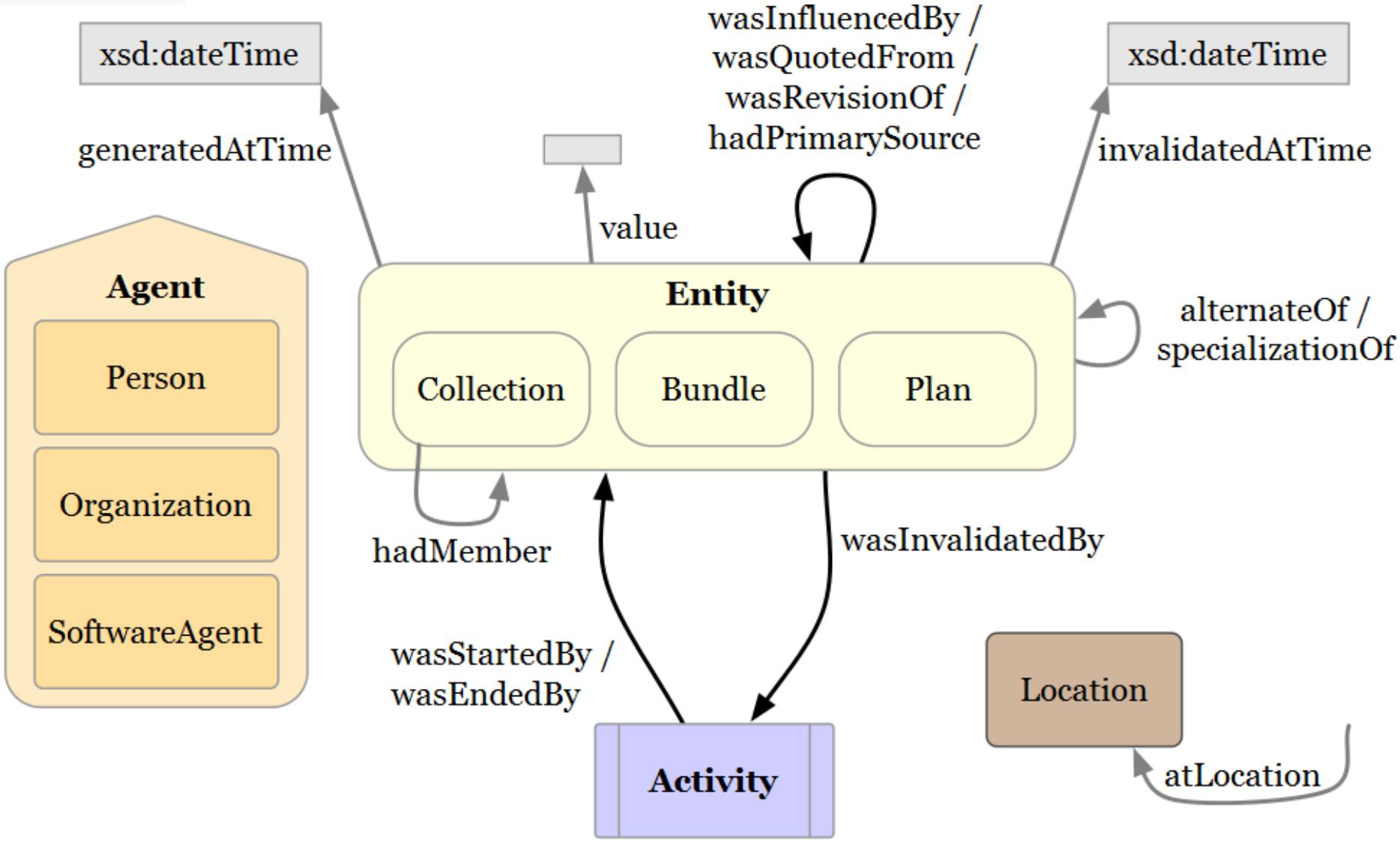
PROV-O: The PROV Ontology Part 2

Part 2

- PROV-O Expanded terms
- PROV-O Qualification terms

PROV-O: Expanded Terms

- Extension of the Starting Point Terms.
- Generic definitions to remain as domain independent as possible.
- Allow for richer descriptions of resources.


xsd:dateTime
generatedAtTime
wasInfluencedBy /
wasQuotedFrom /
wasRevisionOf /
hadPrimarySource
xsd:dateTime
invalidatedAtTime
value
Agent
Person
Organization
SoftwareAgent
Entity
Collection
Bundle
Plan
alternateOf /
specializationOf
hadMember
wasInvalidatedBy
wasStartedBy /
wasEndedBy
Location
Activity
atLocation

- Software Agent, Organization and Person are similar to foaf agents (http://xmlns.com/foaf/spec/#term_Agent).

- These types of Agents are the most common in the Web:
  - Software tools used to create resources.
  - Organizations publishing a document report.
  - Specific authors of a paper.
  - Etc.

"A *SoftwareAgent* is running software".

"An *Organization* is a social or legal institution such as a company, society, etc".

"Person agents are people".

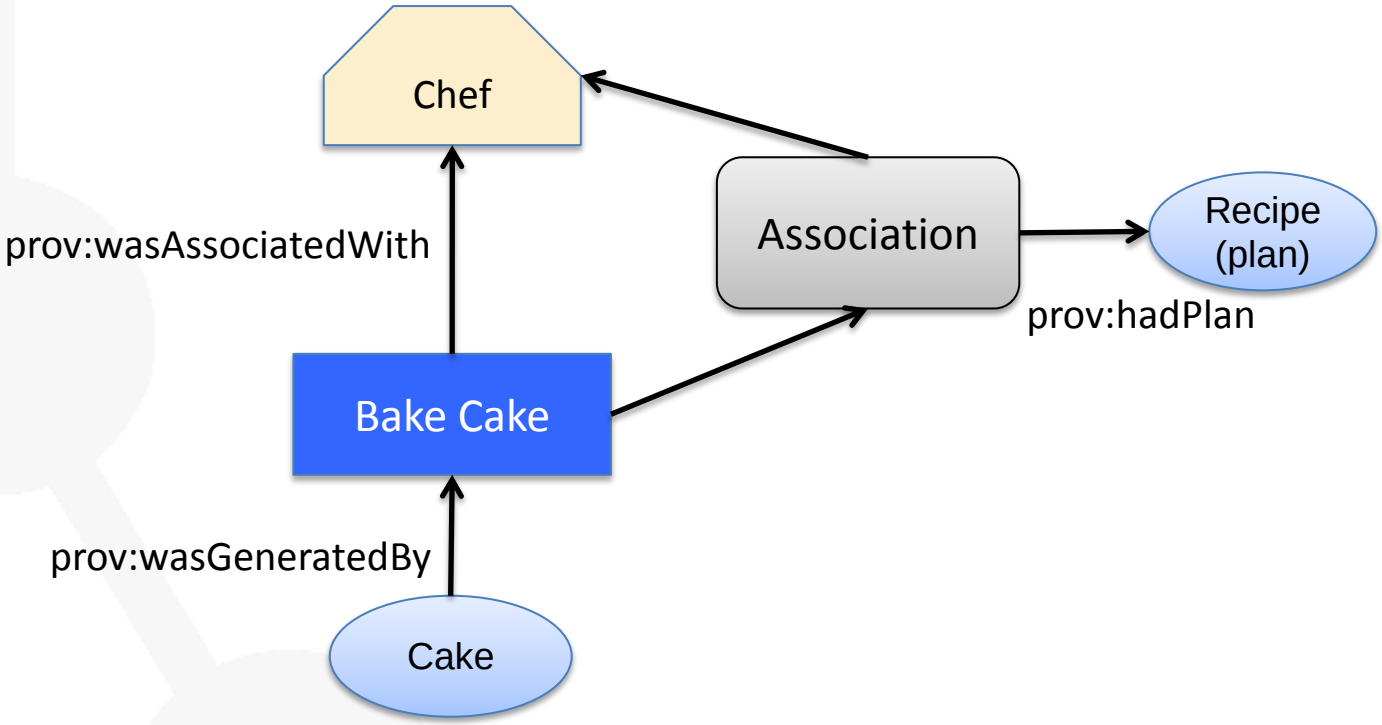
- Locations may be bound to all types of prov resources:
  - Location of a file within a file system.
  - Location where a resource was used or generated.
  - Location where an activity took place.
  - Etc.
- The relationship **prov:atLocation** binds resources to locations

"A *Location* can be an identifiable geographic place (ISO 19112), but it can also be a non-geographic place such as a directory, row, or column. As such, there are numerous ways in which location can be expressed, such as by a coordinate, address, landmark, and so forth".

- A plan can be anything that indicates how to achieve a goal:
  - A script program.
  - A set of instructions written on a napkin.
  - A food recipe.
  - A scientific workflow template.
  - An algorithm.
  - Best practice guidelines.
  - Etc.

"A *plan* is an entity that represents a set of actions or steps intended by one or more agents to achieve some goals".

- Plans are associated to Activities and executed by an Agents:



- There is a complete PROV extension for structured collections (dictionaries) : http://www.w3.org/TR/prov-dictionary/

- Members of a collection are asserted with the **prov:hadMember** relationship.

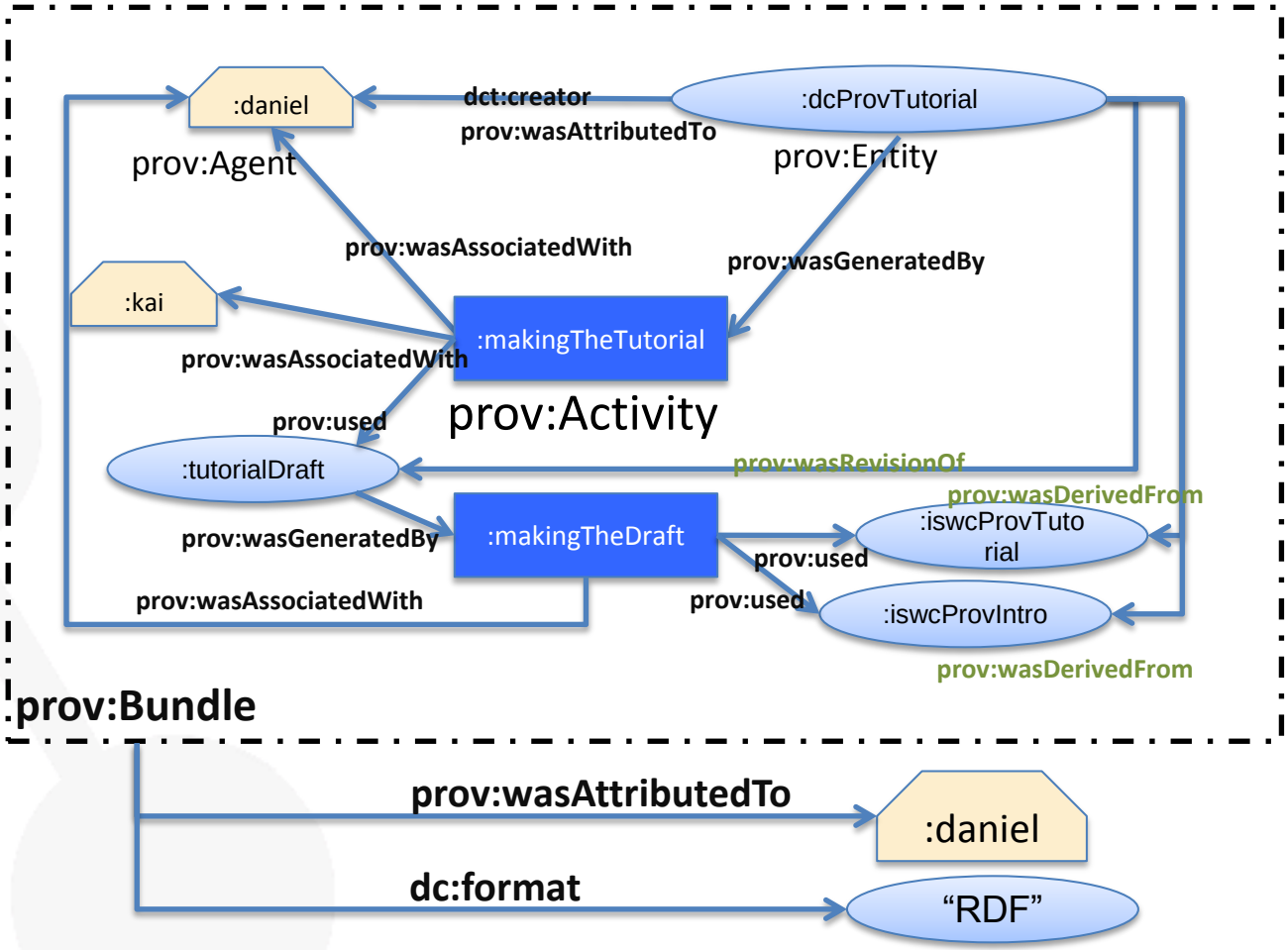
- Examples: the collection of authors who participated in a publication, the members of a research group, etc.

"A *collection* is an entity that provides a structure to some constituents that must themselves be entities. These constituents are said to be *member* of the collections. An *empty collection* is a collection without members".

- Bundles can be any kind of container:
  - Files containing provenance descriptions.
  - Named graphs.
  - Repositories.
  - Etc.
- This document defines how entities could be identified across bundles: http://www.w3.org/TR/prov-links/

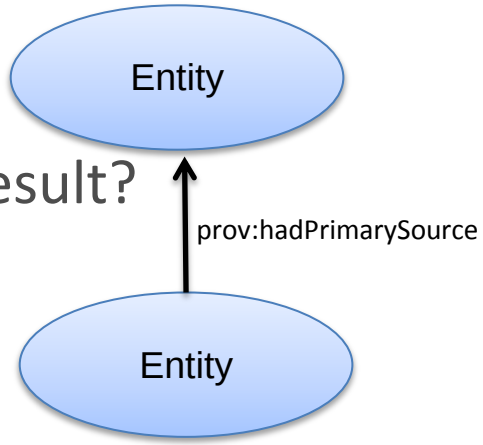
"A *bundle* is a named set of provenance descriptions, and is itself an entity, so allowing provenance of provenance to be expressed."

Describing the provenance of provenance of this presentation (first example)



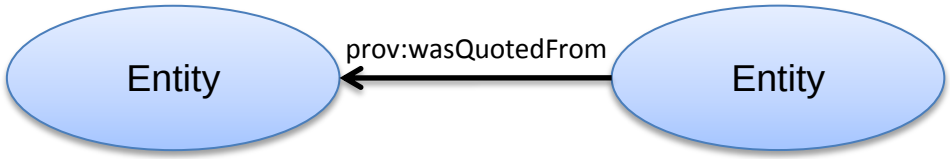
- The determination of primary sources can be up to interpretation.

- Determining the primary source is key for attribution:
  - What is the primary source for a blog post?
  - What is the primary source of a news article?
  - What are the primary sources for a research result?
  - Etc.



"A *primary source* relation is a kind of a derivation relation from secondary materials to their primary sources".

- Quotation is key for giving credit:
  - The quote in this slide was quoted from the prov-dm definition of quotation: http://www.w3.org/TR/prov-dm/#term-quotation
  - Quotes from people in news articles.
  - Etc.
- Quotation is expressed with the relationship prov:wasQuotedFrom.

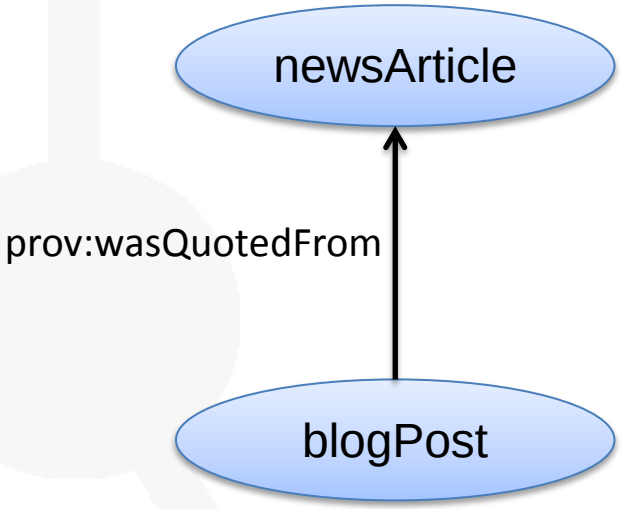


"A *quotation* is the repeat of (some or all of) an entity, such as text or image, by someone who may or may not be its original author".

- Used when entities have a string or numeric value:
  - A string representing the quote in a document.
  - The value of the parameter being used as input for an experimental activity.
  - Etc.

"prov:value provides a value that is a direct representation of an entity".

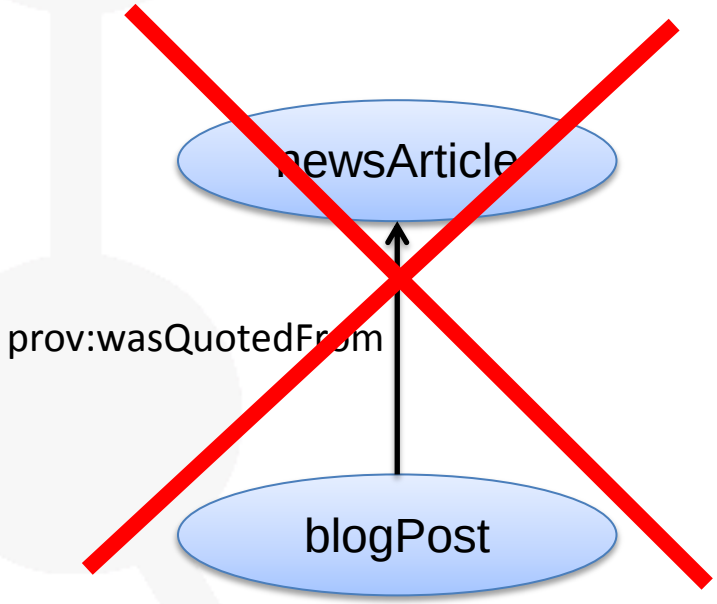
- Quoting a news article in a blog post



"According to the number of registrations in the website, DC2013 conference was a great success"…

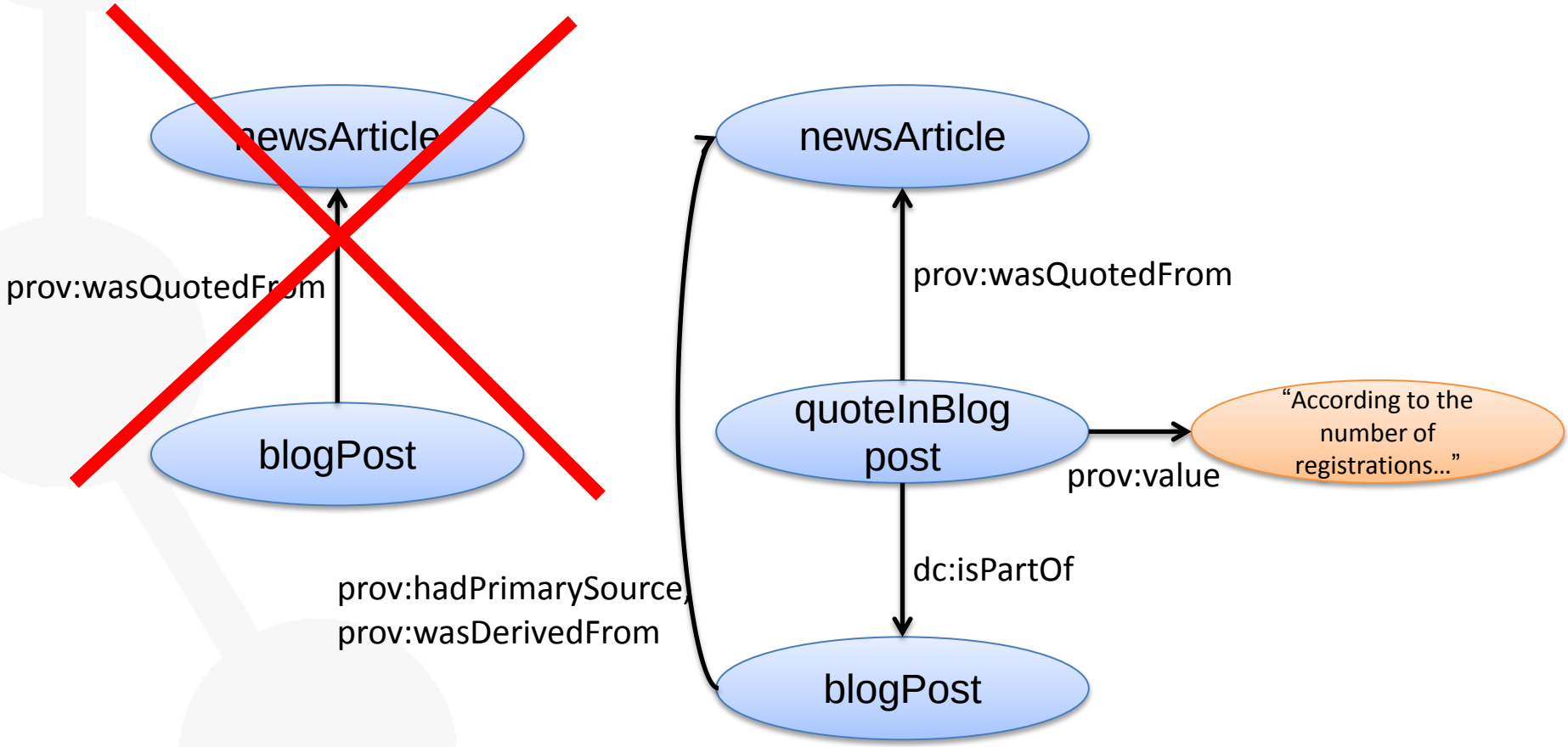
Summary of the DC conference.
As this article states:
"According to the number of registrations in the website, DC2013 conference was a great success"
….

- Quoting a news article in a blog post



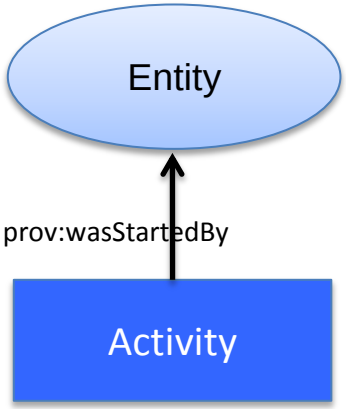
The blog post is not a quote! The wasQuotedFrom relationship should be only for the quotes.

- Quoting a news article in a blog post



- The starter/ender of the activity may be an entity or an agent.

- Useful for assigning responsibility:
  - The cause for the failure of an execution.
  - The email who started a discussion.
  - Etc.

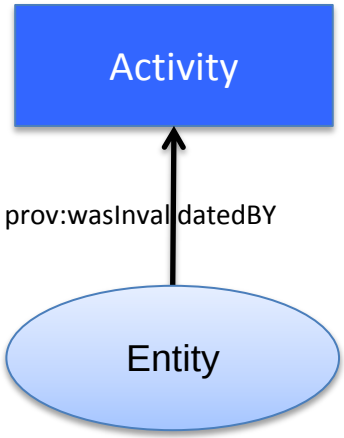


"Start is when an activity is deemed to have been started by an entity, known as *trigger*. The activity did not exist before its start".

"End is when an activity is deemed to have been ended by an entity, known as *trigger*. The activity no longer exists after its end".

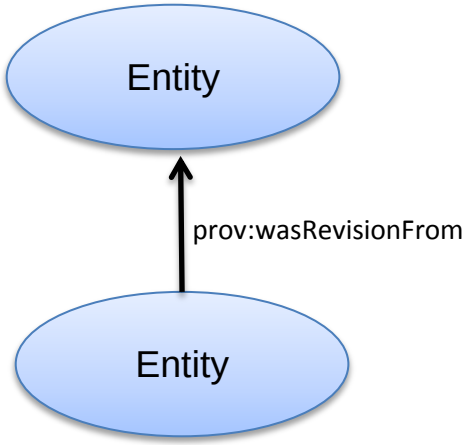
•wasInvalidatedBy is used to point at the activity responsible for the invalidation of the entity:

- A piece of art being damaged by a fire.
- A paper being withdrawn after a discussion with the authors.
- An information invalidated by an automatic process because of its expiry date.
- Etc.



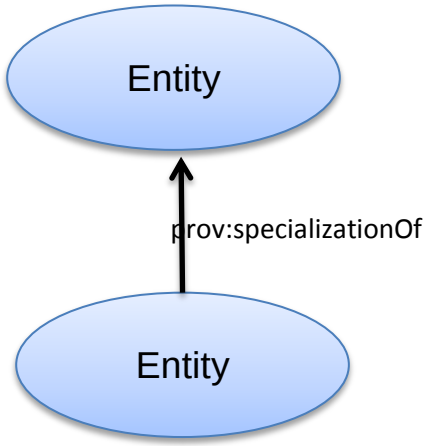
"Invalidation is the start of the destruction, cessation, or expiry of an existing entity by an activity. The entity is no longer available for use (or further invalidation) after invalidation".

- Revision is used for asserting a strong dependency between two entities:
  - A final report is a revision of a previous version.
  - A result is a refinement of previous results.
  - Etc.
- Very similar to dc:isVersionOf.



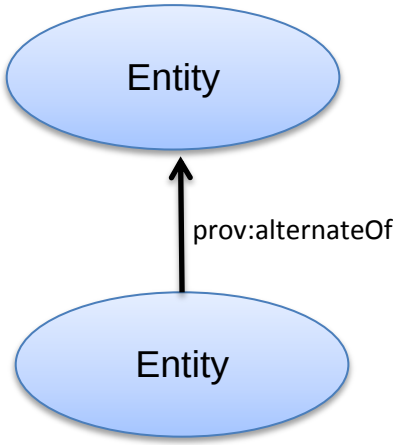
"A *revision* is a derivation for which the resulting entity is a revised version of some original".

- Specialization is used for defining abstractions and contextualized entities:
  - Entities during a period of time (a weather report being a specialization of today's weather report).
  - An agent during a trip is a specialization of that agent.
  - Two different versions of a document can be specializations of the general entity representing the document
  - Etc.



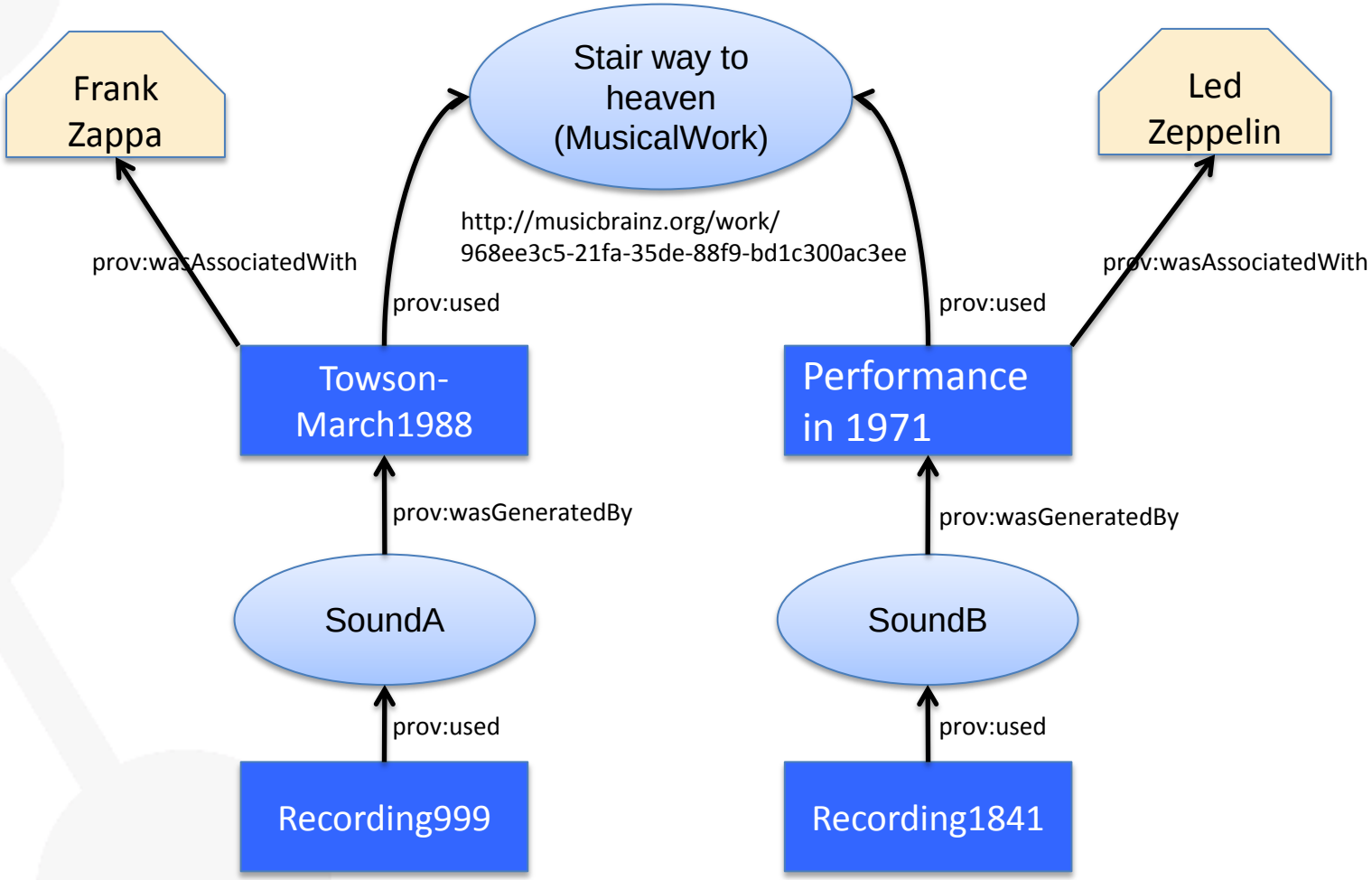
"An entity that is a *specialization* of another shares all aspects of the latter, and additionally presents more specific aspects of the same thing as the latter. In particular, the lifetime of the entity being specialized contains that of any specialization."

- Alternate describe entities that specialize the same resource:
  - Two different versions of a document are alternates of each other.
  - A tool designed for mobile devices is an alternate of a desktop application.
  - Etc.
- Similar to owl:sameAs, but not the same.

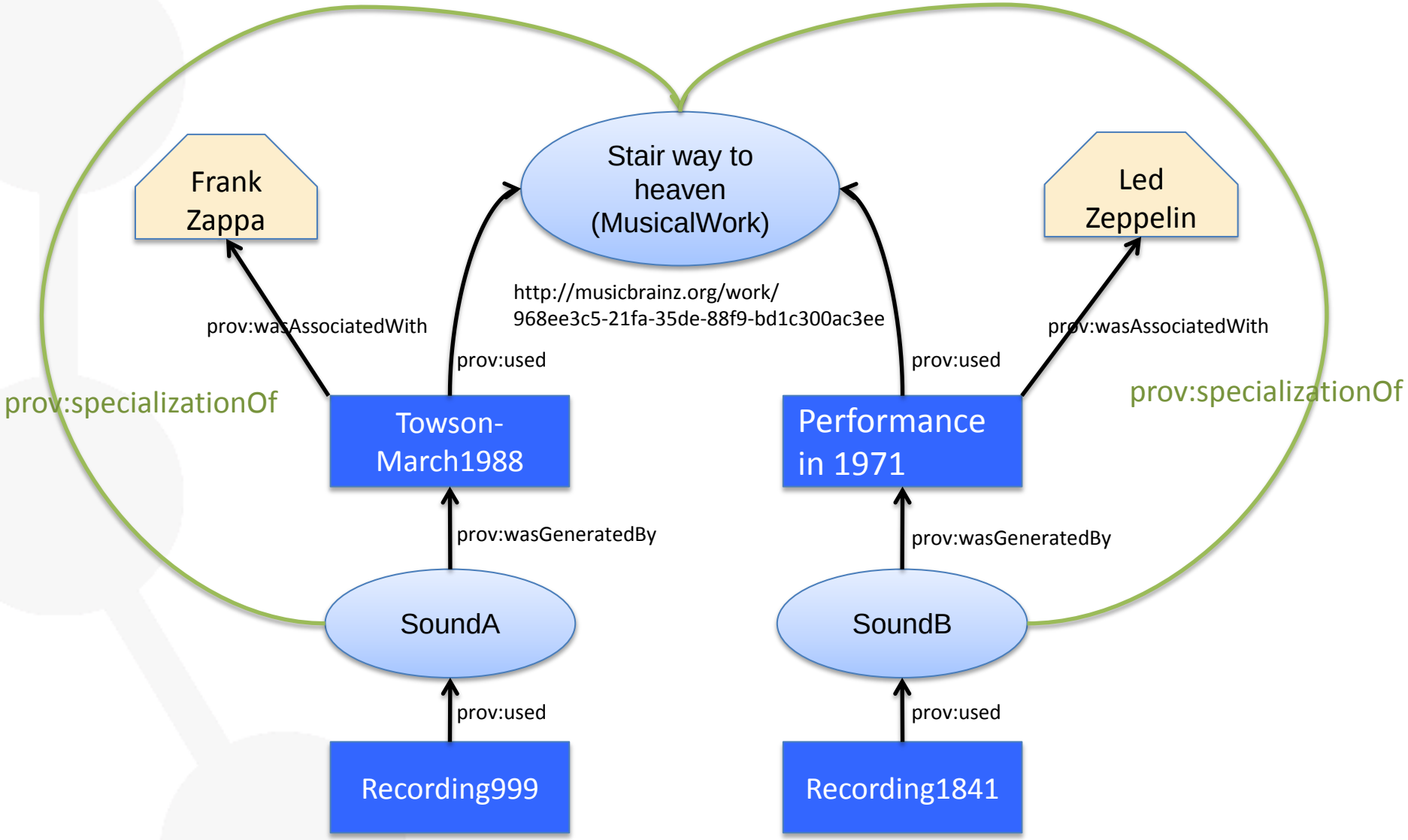


"Two *alternate* entities present aspects of the same thing. These aspects may be the same or different, and the alternate entities may or may not overlap in time."

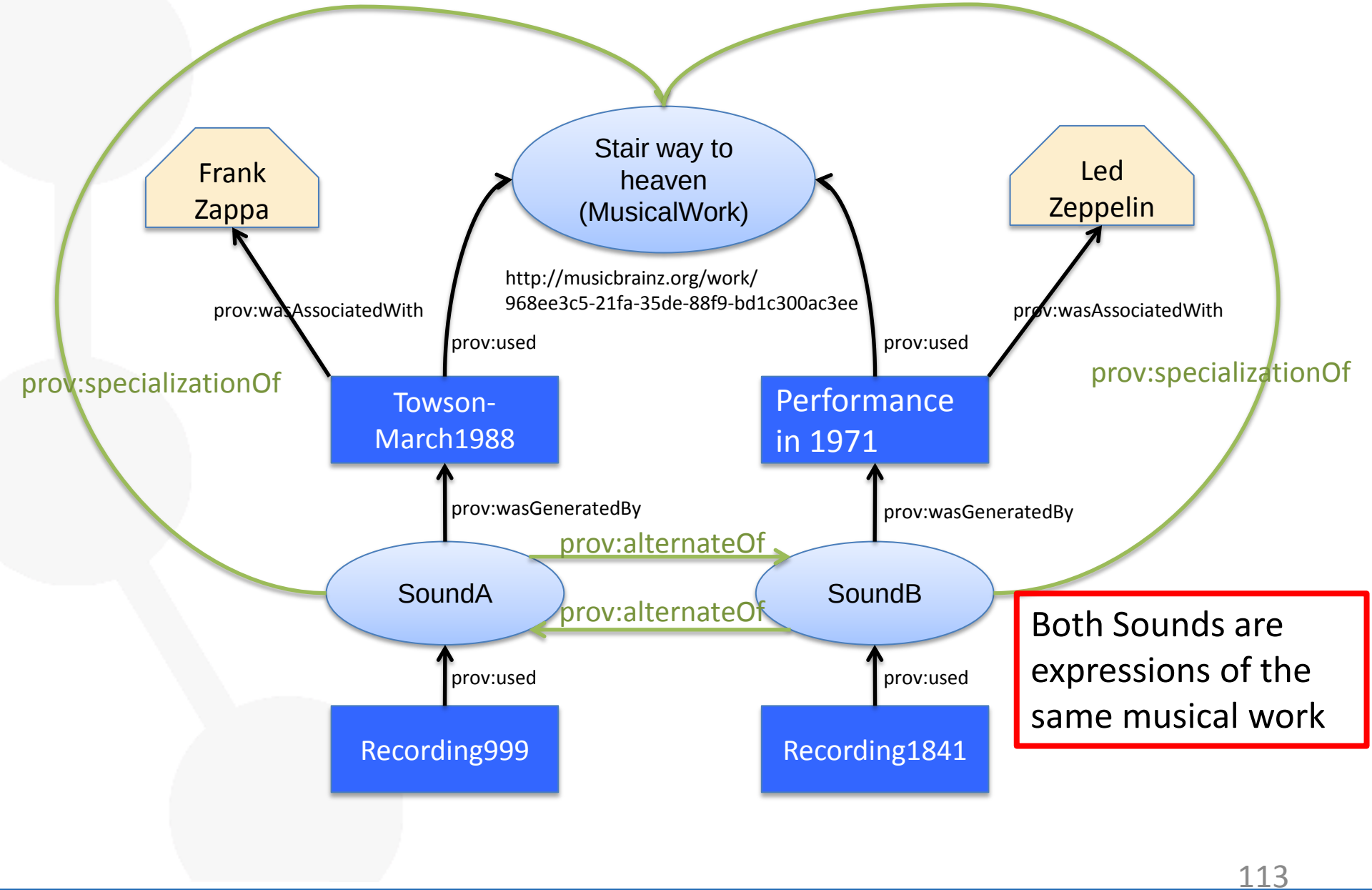
Expanded terms-specialization and alternate: The music example



Expanded terms-specialization and alternate: The music example

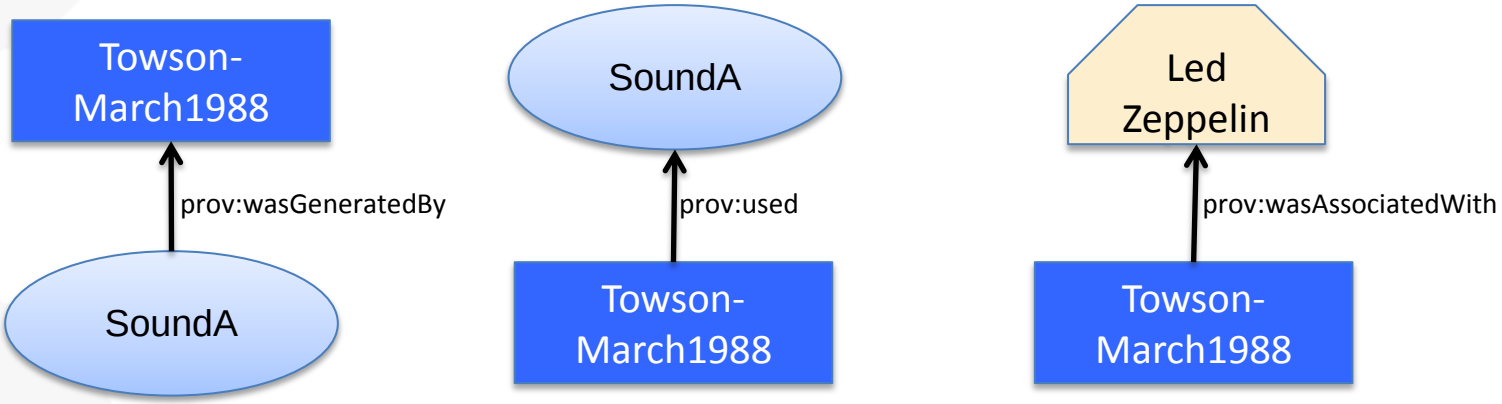


Expanded terms-specialization and alternate: The music example



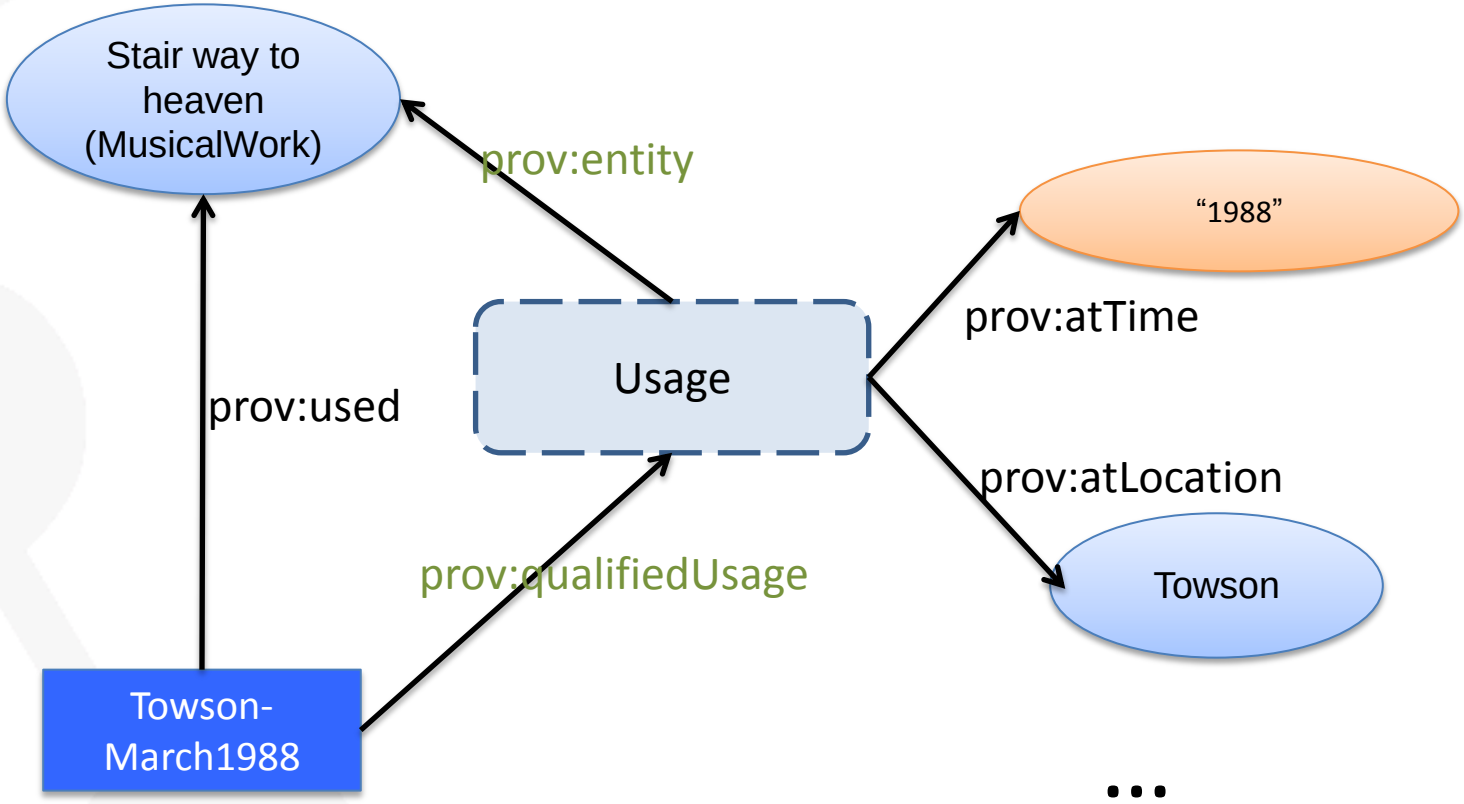
PROV-O: Qualifying terms

- Starting and expanded terms are binary.

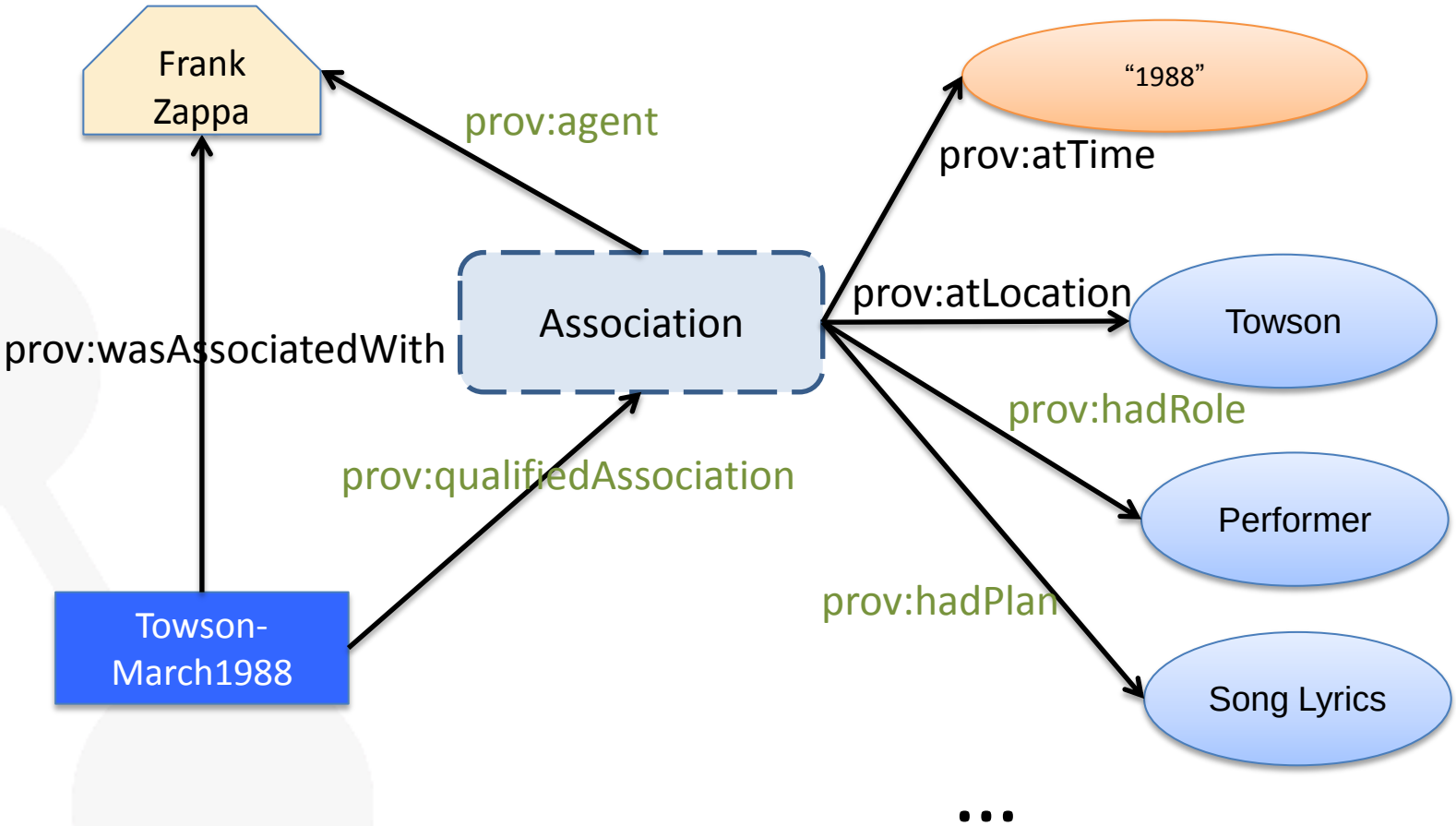


- But we may want to further describe these relationships:
  - Where was an entity generated?
  - What were the roles of the associated to the activity?
  - When was an entity used?
  - Etc.

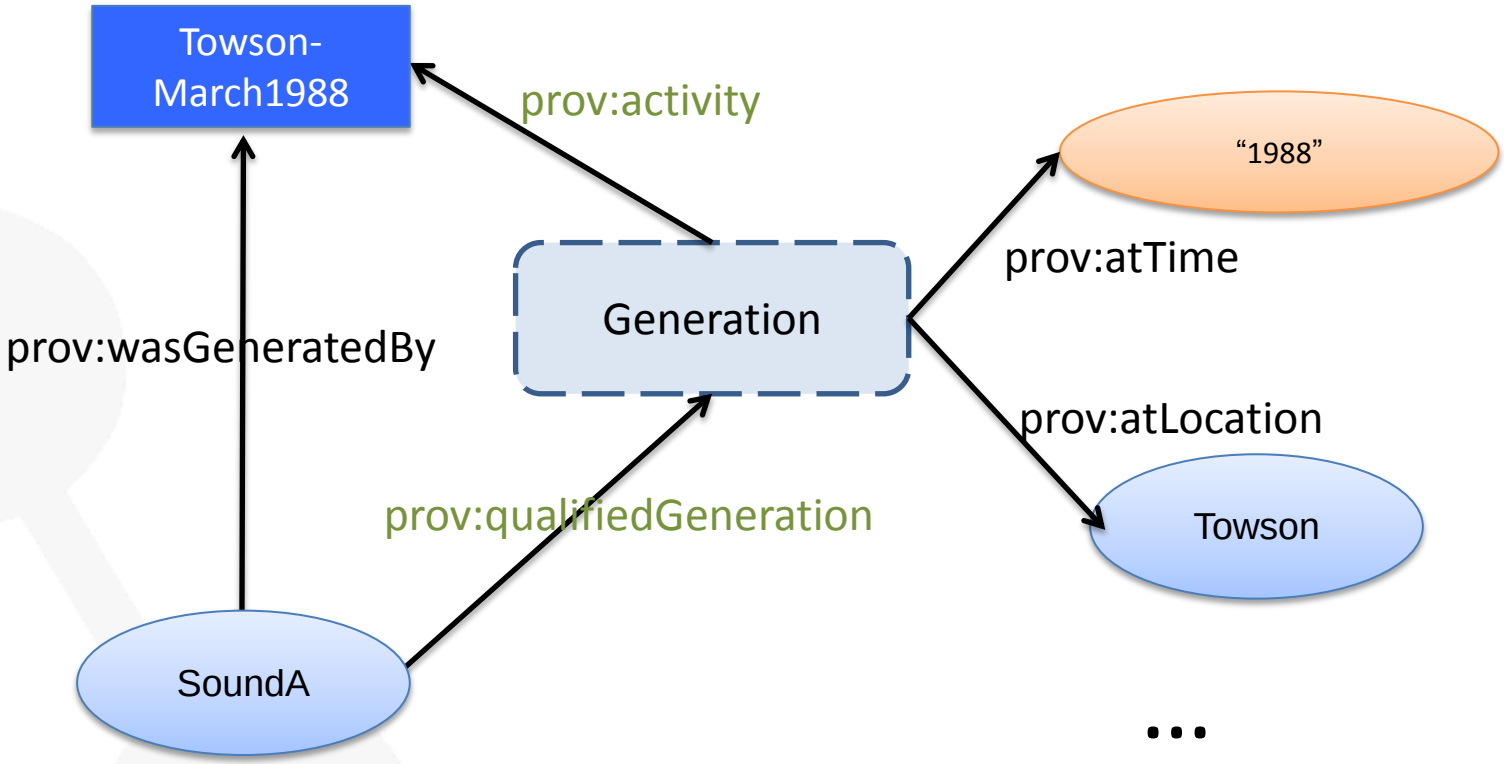
•Qualifying usage:



- Qualifying association:



•Qualifying generation:

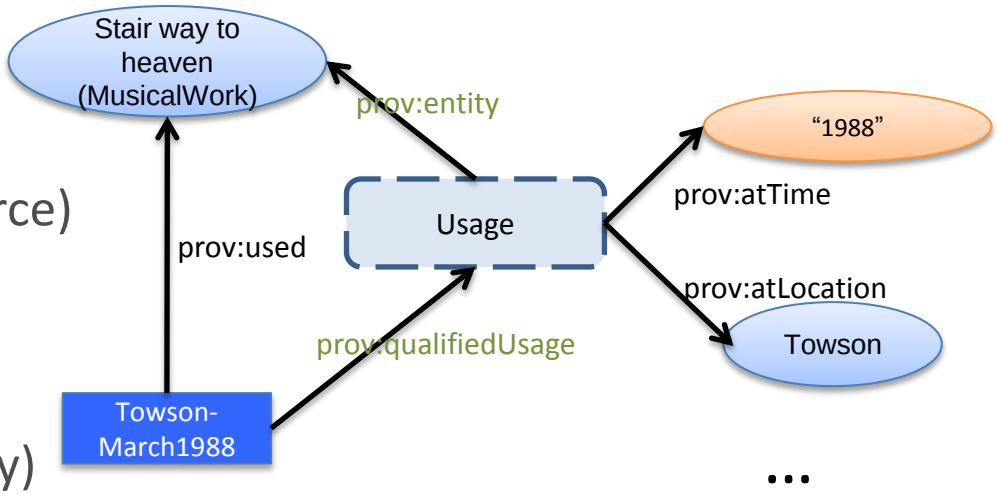


- N-ary relationship pattern:
- http://www.w3.org/TR/swbp-n-aryRelations/

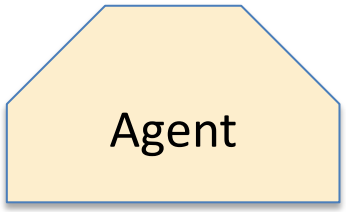
- Example for the usage qualifying pattern:

```
:towsonRecording a prov:Activity;
        prov:used :musicalWork;
        prov:qualifiedUsage [ a prov:Usage;
                prov:entity :musicalWork ;
                prov:hadRole :used-musicalWork;
                prov:atTime “1988-03-23T20:40:40"^^xsd:dateTime;
                prov:atLocation :towson.
        ];
.
:musicalWork a prov:entity;
        dc:title “A stairway to heaven”.
:used-musialWork a prov:Role.
:towson a prov:Location.
```

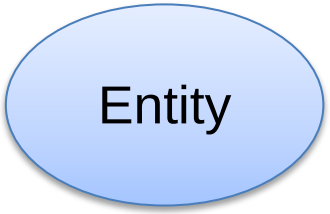
- The following starting point relationships and expanded terms relationships have qualified versions:
  - **Usage (used)**
  - Generation (wasGeneratedBy)
  - Association (wasAssociatedWith)
  - Derivation (wasDerivedFrom)
  - Quotation (wasQuotedFrom)
  - Revision (wasRevisionOf)
  - PrimarySource (hadPrimarySource)
  - Influence (wasInfluencedBy)
  - Start (wasStartedBy)
  - End (wasEndedBy)
  - Communication (wasInformedBy)
  - Invalidation (wasInvalidatedBy)
  - Attribution (wasAttributedTo)
  - Delegation (actedOnBehalfOf)

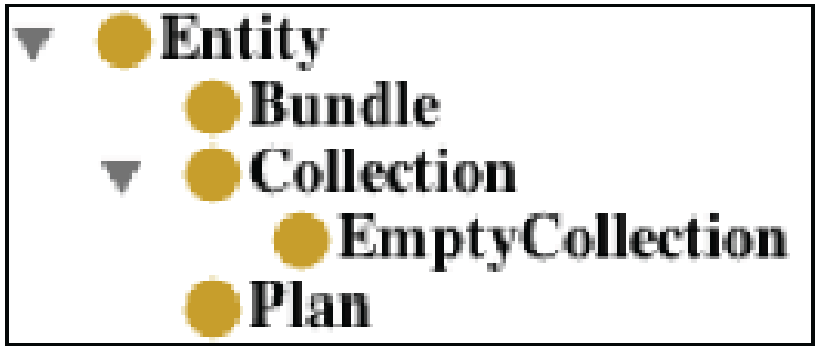


PROV-O: Summary


Agent


Agent
Organization
Person
SoftwareAgent


Entity

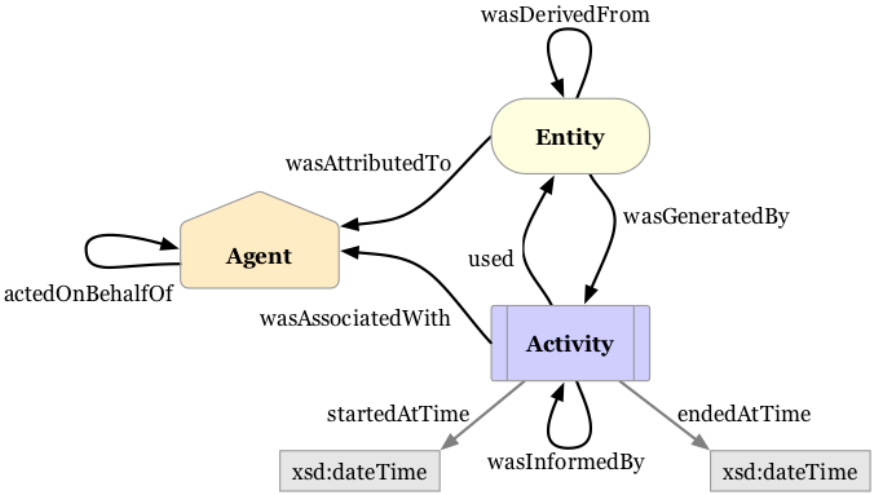

Entity
Bundle
Collection
EmptyCollection
Plan

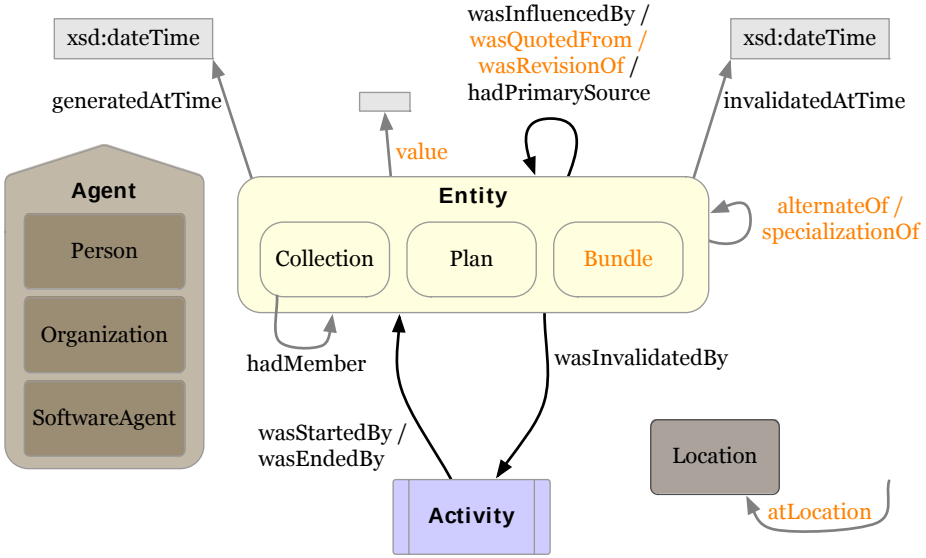

Activity


Activity

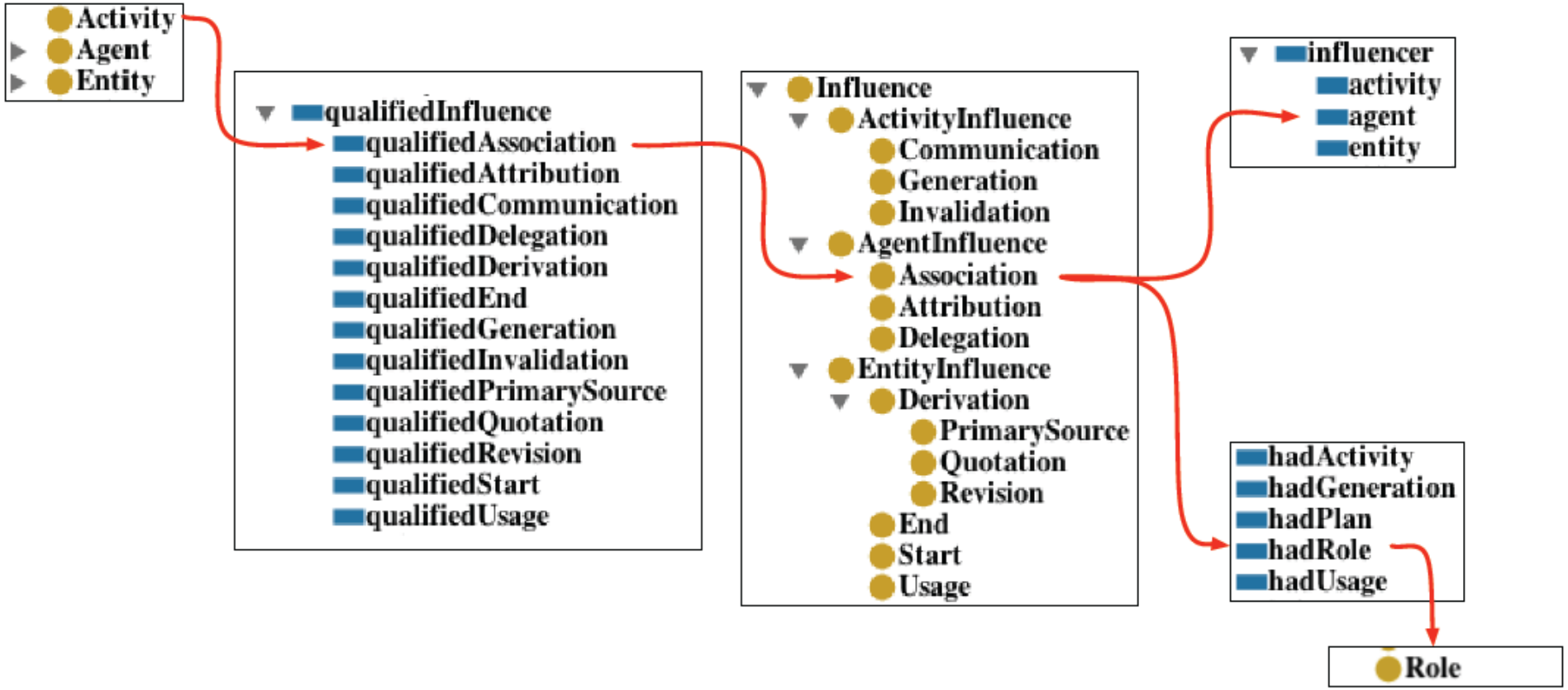
Relating the main 3 classes

- wasInfluencedBy
  - actedOnBehalfOf
  - hadMember
  - used
  - wasAssociatedWith
  - wasAttributedTo
  - wasDerivedFrom
    - hadPrimarySource
    - wasQuotedFrom
    - wasRevisionOf
  - wasEndedBy
  - wasGeneratedBy
  - wasInformedBy
  - wasInvalidatedBy
  - wasStartedBy





Qualifying patterns



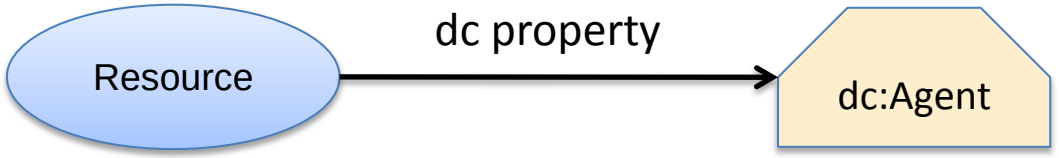
Mapping PROV-O to Dublin Core

http://allfaaraa.com/wp-content/uploads/2013/02/puzzle.jpg

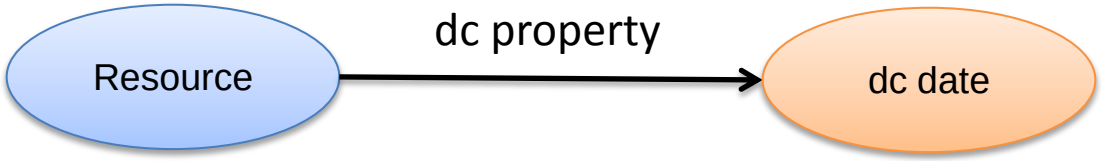
- Preliminaries
  - Provenance and DC
  - Entities in PROV and DC
- Direct mappings
- PROV-O Extensions
- Complex mappings

- Many DC terms hold provenance information
  - Who affected a resource
    - Creator, contributor, publisher, etc..
  - How the resource was affected
    - Access rights, license, hasFormat, etc.
  - When the resource was affected
    - Created, issued, dateSubmitted, etc.
- The rest of the terms hold metadata about the resource
  - Date, description, abstract, language, etc.

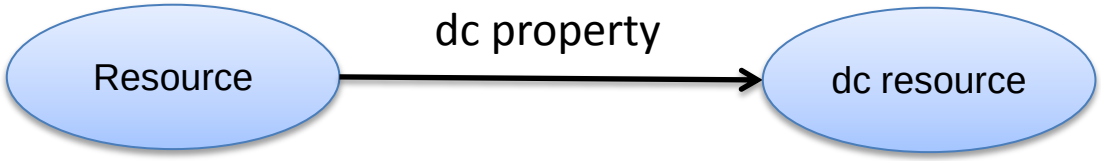
- Properties with dc:Agents as range:
  - creator
  - contributor
  - publisher
  - rightsHolder



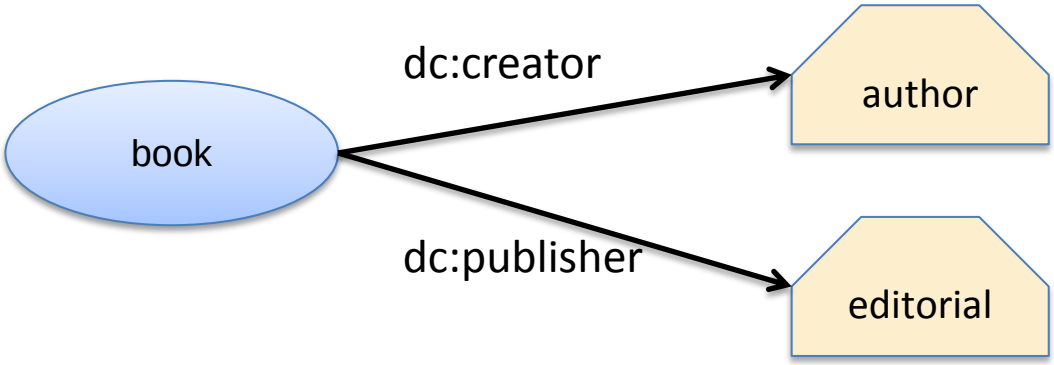
- available
- created
- date
- dateAccepted
- dateCopyrighted
- dateSubmitted
- issued
- modified
- valid



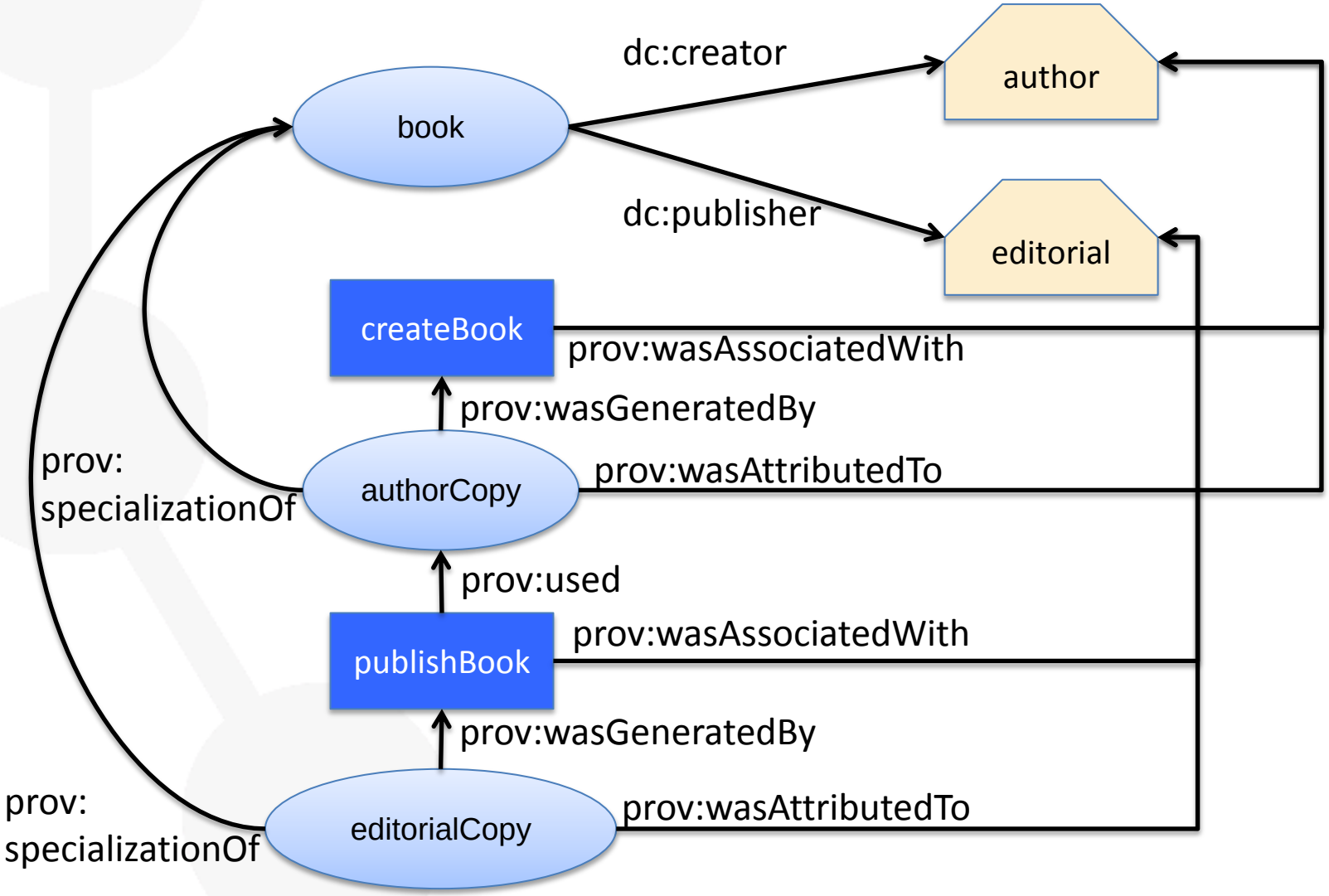
- accessRights
- hasFormat
- hasVersion
- isFormatOf
- isVersionOf
- license
- isReferencedBy
- isReplacedBy
- references
- replaces
- rights
- source



Entities in Dublin Core have “scruffy” provenance…



But in PROV we aim for “complete“ provenance:



- Direct mappings
  - Simple equivalences between PROV terms and DC terms.
  - Described in terms of rdfs:subClassOf, rdfs:subPropertyOf, owl:equivalentClass.
  - The mappings uses prov starting points and prov expanded terms.

Mapping PROV-O to Dublin Core: Direct mappings (terms)

DC Term	Mapping	PROV Property
dateAccepted	subPropertyOf	generatedAtTime
dateAccepted	subPropertyOf	generatedAtTime
dateCopyRighted	subPropertyOf	generatedAtTime
dateSubmitted	subPropertyOf	generatedAtTime
issued	subPropertyOf	generatedAtTime
modified	subPropertyOf	generatedAtTime
creator	subPropertyOf	wasAttributedTo
contributor	subPropertyOf	wasAttributedTo
publisher	subPropertyOf	wasAttributedTo
rightsHolder	subPropertyOf	wasAttributedTo
source	subPropertyOf	wasDerivedFrom
hasFormat	subPropertyOf	alternateOf
isFormatOf	subPropertyOf	alternateOf, wasDerivedFrom

Mapping PROV-O to Dublin Core: Direct mappings (terms)

DC Term	Mapping	PROV Property
dateAccepted	subPropertyOf	generatedAtTime
dateAccepted	subPropertyOf	generatedAtTime
dateCopyRighted	subPropertyOf	generatedAtTime
dateSubmitted	subPropertyOf	generatedAtTime
issued	subPropertyOf	generatedAtTime
modified	subPropertyOf	generatedAtTime
creator	subPropertyOf	wasAttributedTo
contributor	subPropertyOf	wasAttributedTo
publisher	subPropertyOf	wasAttributedTo
rightsHolder	subPropertyOf	wasAttributedTo
source	subPropertyOf	wasDerivedFrom
hasFormat	subPropertyOf	alternateOf
isFormatOf	subPropertyOf	alternateOf, wasDerivedFrom

Generation dates

Mapping PROV-O to Dublin Core: Direct mappings (terms)

DC Term	Mapping	PROV Property
dateAccepted	subPropertyOf	generatedAtTime
dateAccepted	subPropertyOf	generatedAtTime
dateCopyRighted	subPropertyOf	generatedAtTime
dateSubmitted	subPropertyOf	generatedAtTime
issued	subPropertyOf	generatedAtTime
modified	subPropertyOf	generatedAtTime
creator	subPropertyOf	wasAttributedTo
contributor	subPropertyOf	wasAttributedTo
publisher	subPropertyOf	wasAttributedTo
rightsHolder	subPropertyOf	wasAttributedTo
source	subPropertyOf	wasDerivedFrom
hasFormat	subPropertyOf	alternateOf
isFormatOf	subPropertyOf	alternateOf, wasDerivedFrom

Generation dates

Agents

Some DC terms generalize PROV properties:

PROV property	Mapping	DC Term
hadPrimarySource	subPropertyOf	source
wasRevisionOf	subPropertyOf	isVersionOf

Mapping PROV-O to Dublin Core: Main Direct mappings (classes)

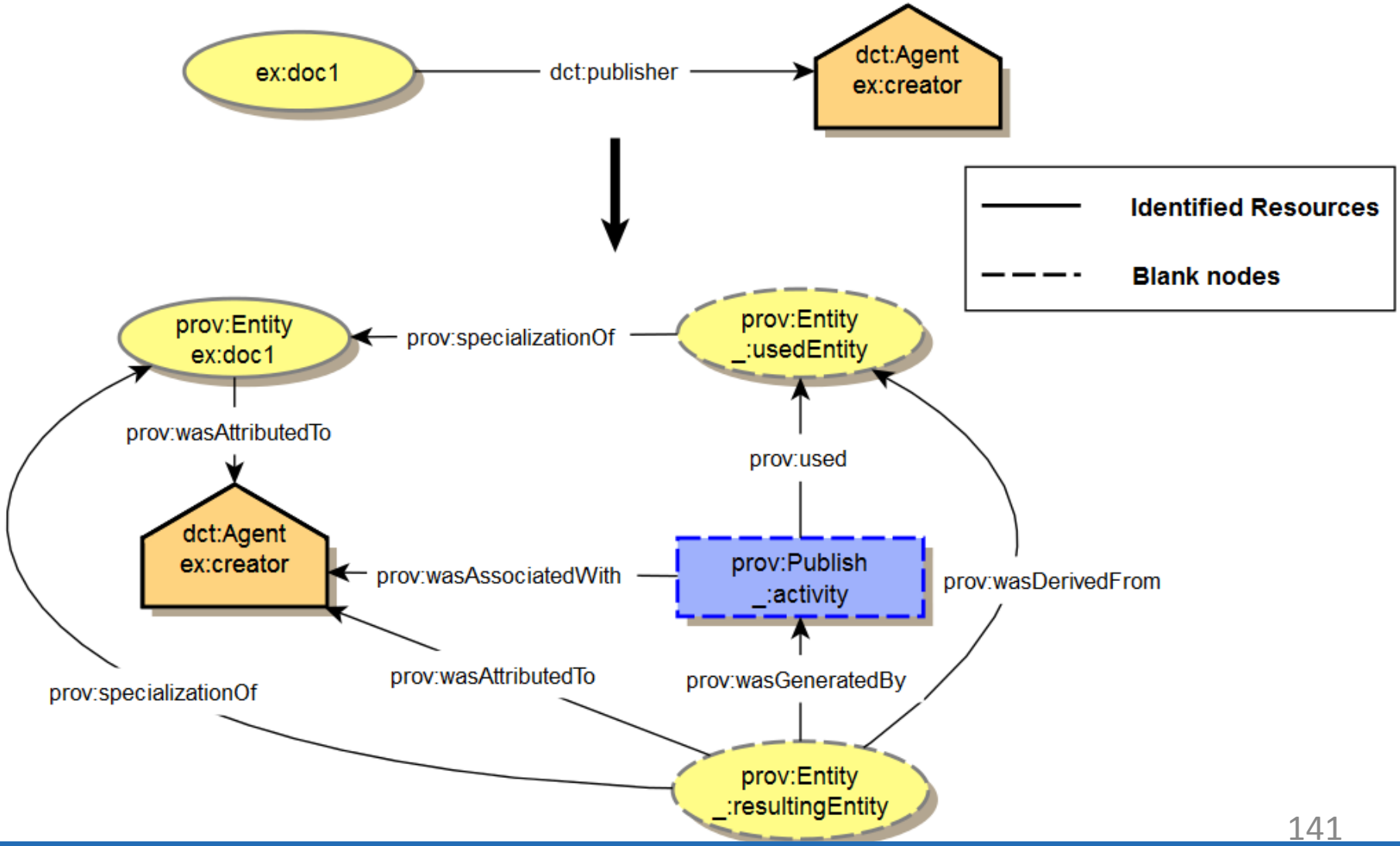
DC Term	Mapping	PROV Property
Agent	equivalentClass	Agent
BibliographicResource	subClassOf	Entity
LicenseDocument	subClassOf	Entity
LinguisticSystem	subClassOf	Plan
Location	equivalentClass	Location
MethodOfAccrual	subClassOf	Plan
MethodOfInstruction	subClassOf	Plan
RightsStatement	subClassOf	Entity
PhysicalResource	subClassOf	Entity
Policy	subClassOf	Plan
ProvenanceStatement	subClassOf	Bundle

To properly represent DC activities and roles, we have extended PROV:

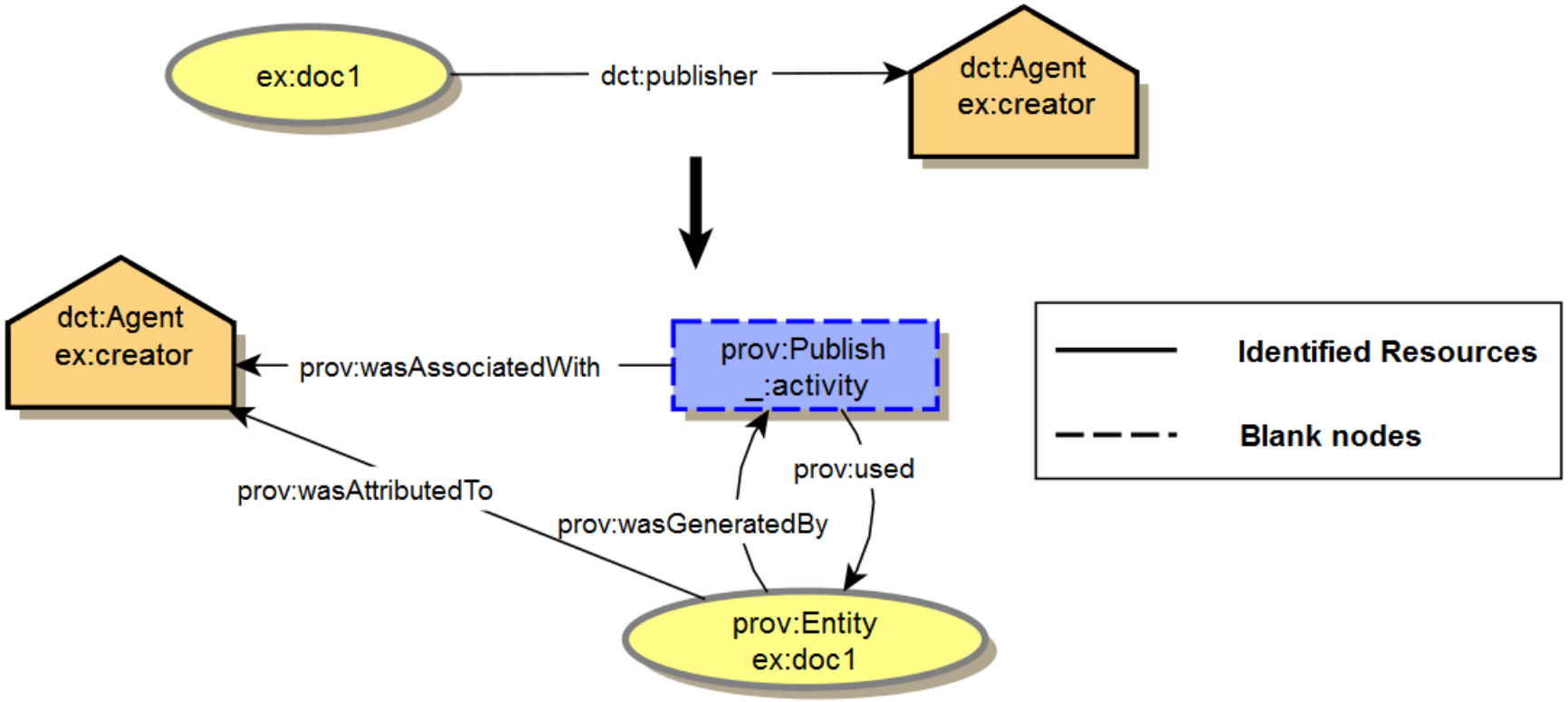
Extended Term	Relation to PROV	PROV extended Term
Publish	subClassOf	Activity
Contribute	subClassOf	Activity
Create	subClassOf	Activity, Contribute
RightsAssignment	subClassOf	Activity
Modify	subClassOf	Activity
Accept	subClassOf	Activity
Copyright	subClassOf	Activity
Submit	subClassOf	Activity
Replace	subClassOf	Activity
Publisher	subClassOf	Role
Contributor	subClassOf	Role
Creator	subClassOf	Role, Contributor
RightsHolder	subClassOf	Role

- Complex mappings
  - Defined to generate qualified PROV statements from DC statements.
  - More complete than simple mappings.
  - May be refined depending on the use case scenario where applied.
  - Provided in the form of SPARQL construct queries.

Transformation of dc:publisher to PROV



Why can't we follow DC's approach?



Publish would generate doc1 and then use it.

- It is not what we want to represent!

Mapping PROV-O to Dublin Core-Complex mappings: an Example

```
CONSTRUCT {
 ?document a prov:Entity;
      prov:wasAttributedTo ?agent.
?agent a prov:Agent.
_:usedEntity a prov:Entity;
      prov:specializationOf ?document.
 _:activity a prov:Activity, prov:Publish;
      prov:used _:usedEntity;
      prov:wasAssociatedWith ?agent;
      prov:qualifiedAssociation [
                  a prov:Association;
                  prov:agent ?agent;
                   prov:hadRole [a prov:Publisher].
       ].
_:resultingEntity a prov:Entity;
      prov:specializationOf ?document;
      prov:wasDerivedFrom _:usedEntity;
      prov:wasGeneratedBy _:activity;
      prov:wasAttributedTo ?agent.
} WHERE { ?document dct:publisher ?agent. }
```

Mapping PROV-O to Dublin Core-Complex mappings: an Example

```
CONSTRUCT {
 ?document a prov:Entity;    (GENERAL ENTITY)
      prov:wasAttributedTo ?agent.  (DIRECT MAPPING)
?agent a prov:Agent.
_:usedEntity a prov:Entity;
      prov:specializationOf ?document.
 _:activity a prov:Activity, prov:Publish;
      prov:used _:usedEntity;
      prov:wasAssociatedWith ?agent;
      prov:qualifiedAssociation [
                  a prov:Association;
                  prov:agent ?agent;
                   prov:hadRole [a prov:Publisher].
       ].
_:resultingEntity a prov:Entity;
      prov:specializationOf ?document;
      prov:wasDerivedFrom _:usedEntity;
      prov:wasGeneratedBy _:activity;
      prov:wasAttributedTo ?agent.
} WHERE { ?document dct:publisher ?agent. }
```

Mapping PROV-O to Dublin Core-Complex mappings: an Example

```
CONSTRUCT {
 ?document a prov:Entity;    (GENERAL ENTITY)
      prov:wasAttributedTo ?agent.  (DIRECT MAPPING)
?agent a prov:Agent.
_:usedEntity a prov:Entity;  (SPECIALIZATION OF THE GENERAL  ENTITY)
      prov:specializationOf ?document.
 _:activity a prov:Activity, prov:Publish;
      prov:used _:usedEntity;
      prov:wasAssociatedWith ?agent;
      prov:qualifiedAssociation [
                  a prov:Association;
                  prov:agent ?agent;
                   prov:hadRole [a prov:Publisher].
       ].
_:resultingEntity a prov:Entity; (SPECIALIZATION OF THE GENERAL  ENTITY)
      prov:specializationOf ?document;
      prov:wasDerivedFrom _:usedEntity;
      prov:wasGeneratedBy _:activity;
      prov:wasAttributedTo ?agent.
} WHERE { ?document dct:publisher ?agent. }
```

Mapping PROV-O to Dublin Core-Complex mappings: an Example

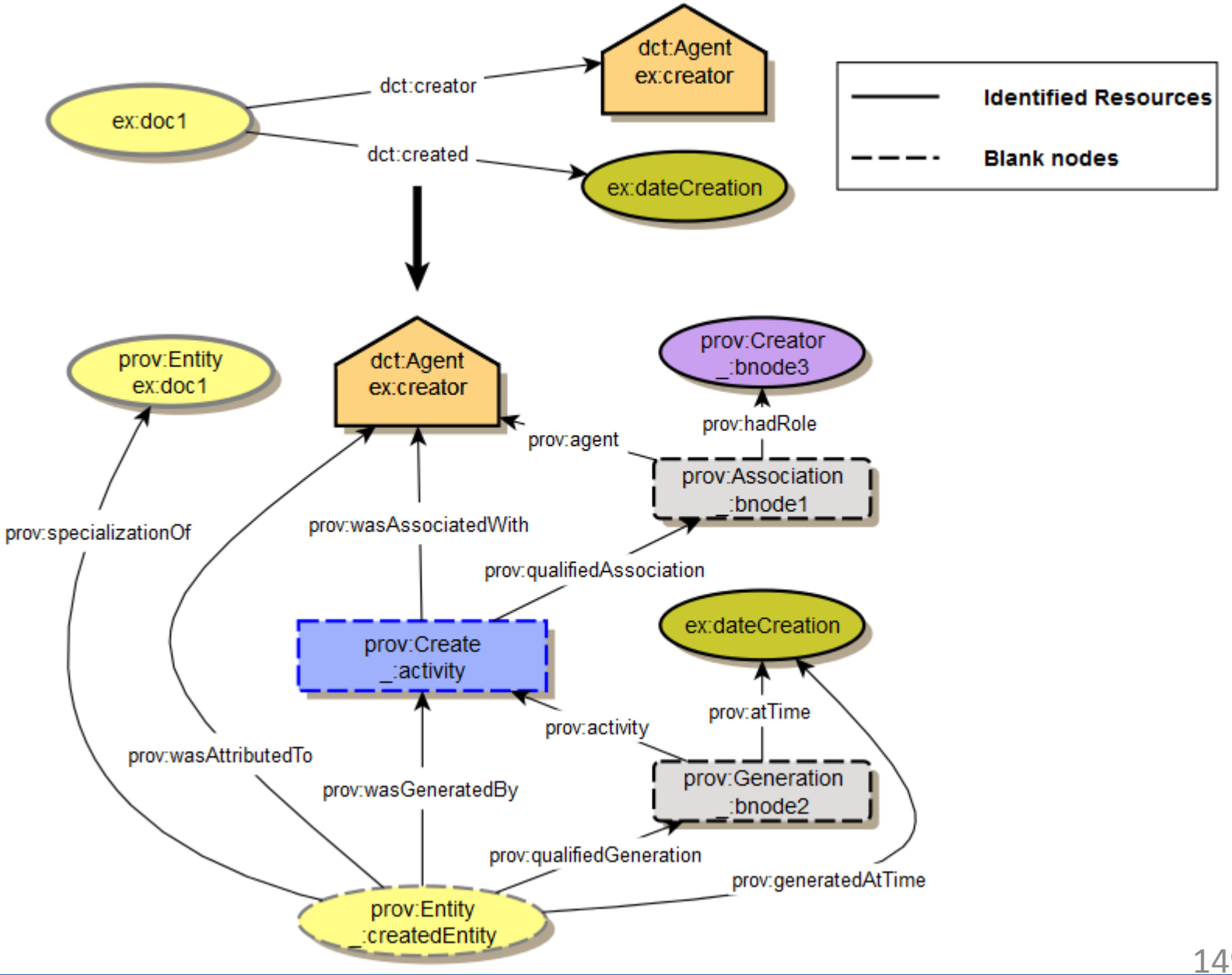
```
CONSTRUCT {
 ?document a prov:Entity;    (GENERAL ENTITY)
      prov:wasAttributedTo ?agent.  (DIRECT MAPPING)
?agent a prov:Agent.
_:usedEntity a prov:Entity;  (SPECIALIZATION OF THE GENERAL  ENTITY)
      prov:specializationOf ?document.
 _:activity a prov:Activity, prov:Publish; (ACTIVITY EXTENDING PROV)
      prov:used _:usedEntity;
      prov:wasAssociatedWith ?agent;
      prov:qualifiedAssociation [
                  a prov:Association;
                  prov:agent ?agent;
                   prov:hadRole [a prov:Publisher].
       ].
_:resultingEntity a prov:Entity; (SPECIALIZATION OF THE GENERAL  ENTITY)
      prov:specializationOf ?document;
      prov:wasDerivedFrom _:usedEntity;
      prov:wasGeneratedBy _:activity;
      prov:wasAttributedTo ?agent.
} WHERE { ?document dct:publisher ?agent. }
```

Mapping PROV-O to Dublin Core-Complex mappings: an Example

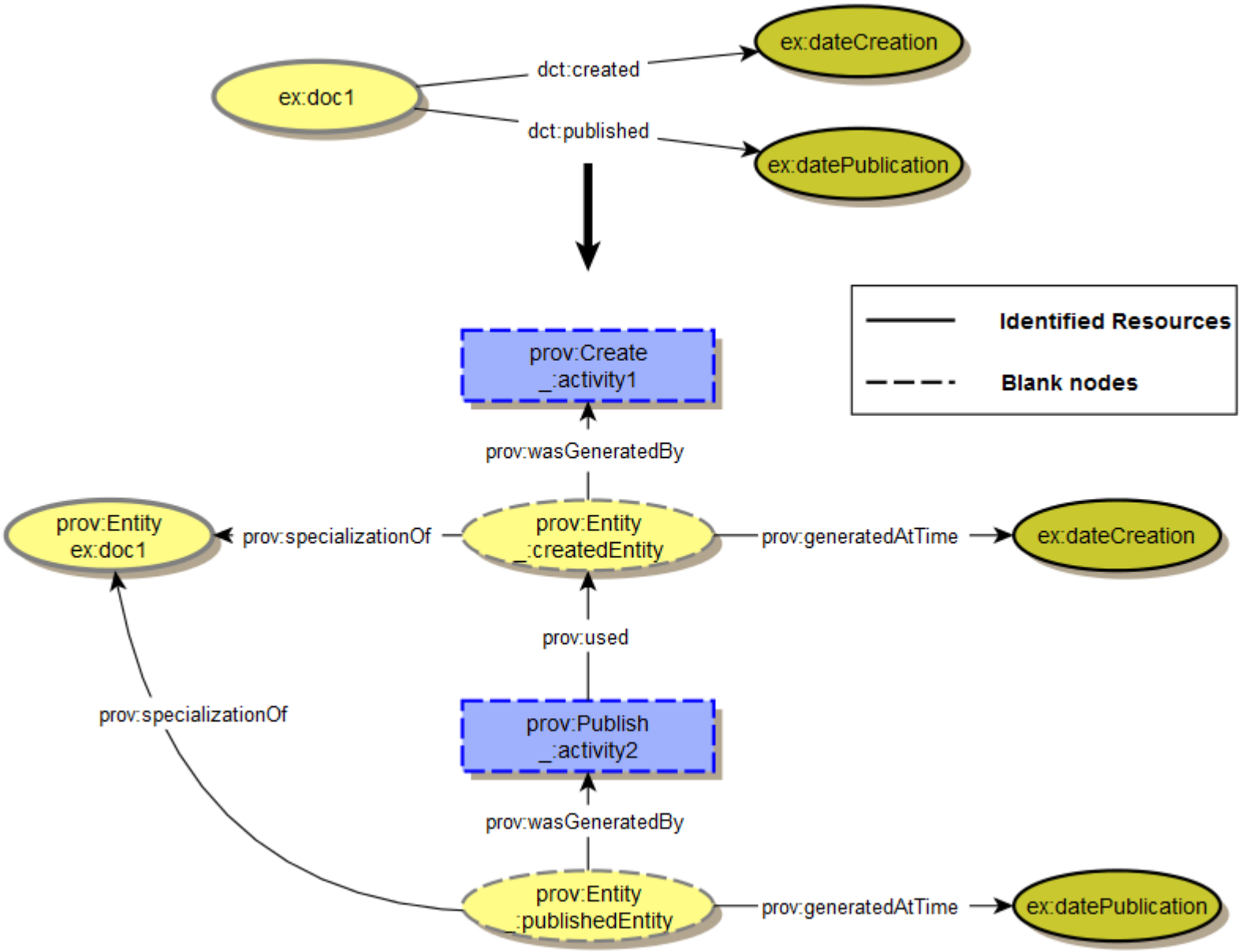
```
CONSTRUCT {
 ?document a prov:Entity;    (GENERAL ENTITY)
      prov:wasAttributedTo ?agent.  (DIRECT MAPPING)
?agent a prov:Agent.
_:usedEntity a prov:Entity;  (SPECIALIZATION OF THE GENERAL  ENTITY)
      prov:specializationOf ?document.
 _:activity a prov:Activity, prov:Publish; (ACTIVITY EXTENDING PROV)
      prov:used _:usedEntity;
      prov:wasAssociatedWith ?agent;
      prov:qualifiedAssociation [ (QUALIFIED ASSOCIATION TO BIND THE ACTIVITY TO THE ROLE)
                  a prov:Association;
                  prov:agent ?agent;
                   prov:hadRole [a prov:Publisher].
       ].
_:resultingEntity a prov:Entity; (SPECIALIZATION OF THE GENERAL  ENTITY)
      prov:specializationOf ?document;
      prov:wasDerivedFrom _:usedEntity;
      prov:wasGeneratedBy _:activity;
      prov:wasAttributedTo ?agent.
} WHERE { ?document dct:publisher ?agent. }
```

- The complex mapping lead to the proliferation of blank nodes.
  - Providing URIs for the blank nodes may solve the problem
  - Is there a way to reduce the number of blank nodes?
    - Conflate properties referring to the same state of the resource
    - Sort activities by their logical order

Mapping PROV-O to Dublin Core: Cleanup 1



Mapping PROV-O to Dublin Core: Cleanup 2



- Direct mappings
- PROV-O Extensions
- Complex mappings

- Luc Moreau, Ivan Herman, Paul Groth and Timothy Lebo for creating some of the materials used for this presentation:
  - http://www.w3.org/2001/sw/wiki/OutreachInformation

- Victor Rodriguez Doncel for providing the sources to the music ontology example.

- Kai Eckert, María Poveda, Oscar Corcho and Daniel Nüst for providing feedback.

- PROV-O team: http://www.w3.org/2011/prov#prov-o-team

Questions?

http://thumbs.dreamstime.com/x/question-mark-five-w-s-18903552.jpg

PROV-O: The PROV Ontology Tutorial

Daniel Garijo

Ontology Engineering Group
Universidad Politécnica de Madrid

(with Slides from Luc Moreau, Ivan Herman, Paul Groth and Timothy Lebo)

Date: 01/09/2013